NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

KHAFJI:
A COMBAT SIMULATION

by

Malcolm W. Garland

September, 1991

Thesis Advisor: Samuel H. Parry

Approved for public release; distribution is unlimited

Unclassified

security classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified		1b Restrictive Markings	
2a Security Classification Authority		3 Distribution Availability of Report Approved for public release; distribution is unlimited.	
2b Declassification Downgrading Schedule			
4 Performing Organization Report Number(s)		5 Monitoring Organization Report Number(s)	
6a Name of Performing Organization Naval Postgraduate School	6b Office Symbol *(if applicable)* 30	7a Name of Monitoring Organization Naval Postgraduate School	
6c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000		7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000	
8a Name of Funding Sponsoring Organization	8b Office Symbol *(if applicable)*	9 Procurement Instrument Identification Number	
8c Address *(city, state, and ZIP code)*		10 Source of Funding Numbers	
		Program Element No / Project No	Task No / Work Unit Accession No
11 Title *(include security classification)* KHAFJI: A COMBAT SIMULATION			
12 Personal Author(s) Malcolm W. Garland			
13a Type of Report Master's Thesis	13b Time Covered From To	14 Date of Report *(year, month, day)* September 1991	15 Page Count 137
16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
17 Cosati Codes: Field / Group / Subgroup		18 Subject Terms *(continue on reverse if necessary and identify by block number)* mine, minefield, SIMSCRIPT, SIMGRAPHICS, breach, hasty breach.	

19 Abstract *(continue on reverse if necessary and identify by block number)*

This thesis presents a high resolution, discrete event driven combat simulation. This model was developed to facilitate the analysis of tactical options available to a small unit (company/platoon) commander using artillery and multiple lanes in overcoming a minefield obstacle.

KHAFJI is a high fidelity combat simulation written in SIMSCRIPT II.5 with SIMGRAPHICS I. Employing user input parameters which define a minefield scenario, the model generates output which enables the user to compare various tactical options available to a maneuver commander in crossing a minefield. By using menu driven input screens, the user has a choice of multiple crossing lanes, size of crossing force, distribution of forces upon crossing lanes, multiple mine belts, and use of indirect fires against the minefield.

Using SIMGRAPHICS I software, KHAFJI displays the minefield and the unit as it crosses the minefield. KHAFJI depicts each mine, each member of the crossing unit, and each impacting artillery round. The graphics provided by KHAFJI allows the user to see the crossing as it unfolds, thereby, reinforcing user confidence in the resultant data. When running multiple replications, graphics can be turned off to speed processing.

An example of the type of analysis that can be performed with KHAFJI is presented in Chapter IV.

20 Distribution Availability of Abstract ☒ unclassified unlimited ☐ same as report ☐ DTIC users	21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual Samuel Parry	22b Telephone *(include Area code)* (408) 646-2787	22c Office Symbol 30

DD FORM 1473,84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete — security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

KHAFJI:
A COMBAT SIMULATION

by

Malcolm W. Garland
Captain, United States Army
B.S., University of North Carolina at Greensboro, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1991

Peter Purdue, Chairman,
Department of Operations Researh

ABSTRACT

This thesis presents a high resolution, discrete event driven combat simulation. This model was developed to facilitate the analysis of tactical options available to a small unit (company/platoon) commander using artillery and multiple lanes in overcoming a minefield obstacle.

KHAFJI is a high fidelity combat simulation written in SIMSCRIPT II.5 with SIMGRAPHICS I. Employing user input parameters which define a minefield scenario, the model generates output which enables the user to compare various tactical options available to a maneuver commander in crossing a minefield. By using menu driven input screens, the user has a choice of multiple crossing lanes, size of crossing force, distribution of forces upon crossing lanes, multiple mine belts, and use of indirect fires against the minefield.

Using SIMGRAPHICS I software, KHAFJI displays the minefield and the unit as it crosses the minefield. KHAFJI depicts each mine, each member of the crossing unit, and each impacting artillery round. The graphics provided by KHAFJI allows the user to see the crossing as it unfolds, thereby, reinforcing user confidence in the resultant data. When running multiple replications, graphics can be turned off to speed processing.

An example of the type of analysis that can be performed with KHAFJI is presented in Chapter IV.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. HISTORICAL PROSPECTIVE

It was a hot, dry, listless day of October, 1942. In a remote command post in a desolate Egyptian desert strutted an exuberant, jubilantly confident commander. Throughout the previous season, he had been defeated, his forces demoralized, his nation humiliated. Now his time had come. This time he would win. His forces outnumbered his opponent's forces by over 3 to 1 in tanks and artillery. His force was well armed, well trained, and well rested. He was on the offensive, he had the initiative, he could not be stopped, and he knew it. Therefore, he decided to sleep through the initial phases of the battle. [Ref. 1]

In another command post sat a frustrated, disappointed, and dejected commander. His forte up to this point had been lightening maneuver but now he was outnumbered 3 to 1 in tanks and artillery, his ammunition was nearly depleted, his petrol was spent, and his supply lines were shattered. He had lost the initiative. He was about to be attacked, he was about to be defeated, and he knew it. But he also knew that he must act to save his force from annihilation. His name was Erwin Rommel, his nemesis was Bernard Montgomery. [Ref. 2]

Rommel's decision was to lay 500,000 landmines. His force, badly mauled, survived to fight on. [Ref. 3]

Forty years later in October 1991, in an Arabian desert, sat a commander who had taken the first step in restoring the prestige of his nation. Some years earlier a sovereign state had been carved out of the southeastern province of his nation. He had erased the humiliation by restoring his God given frontier. Now his enemies had gathered a mighty coalition force against him. He was about to be attacked and he knew it. But he had to act to save his forces from the impending coalition onslaught. His name was Sadam Hussein. His decision was to lay 1,000,000 landmines on the Saudi Arabian-Kuwaiti frontier.

B. PROBLEM

The Iraqi defensive barrier facing Desert Storm forces was formidable. The obstacle network consisted of three densely laid mine belts (approximately 100 meters in depth), reinforced by oil filled tank ditch networks (approximately 50 meters in depth). The defensive scenario envisioned that as a force penetrated the first mine belt, the force then

must negotiate a flaming tank ditch network (oil set ablaze). Once across this tank ditch network, the cycle must be repeated twice more before the attacking unit could clear the obstacle. In addition to the mine tank ditch obstacles, the entire defensive network was to be covered by enemy covering fires. (See Figure 1 on page 3).[Ref. 4]

This defensive barrier presented an acute problem for Desert Storm operational planners. This thesis concerns the minefield aspect of this problem.

Landmines are effective counter-mobility weapons--they slow or halt armored columns, they cause casualties, and they are cheap. Today, with the advent of anti-shock, non--metallic, anti-armor, anti-disturbance landmines, mine warfare has become a potentially devastating combat multiplier. While mine warfare has enjoyed these technological advances, counter-mine warfare has lagged behind. Since WWII, the U.S. Army's only deployed technological solution to this problem has been detonating line charges, whose effectiveness against anti-shock landmines is suspect [Ref. 5].

In the absence of mine defeating technologies, tactics must be employed to overcome the existing mine/counter-mine warfare technological gap. This presents a question: What is the "best" tactic for moving through a minefield?

C. PURPOSE

The purpose of this thesis is develop a high resolution simulation to analyze tactics which use artillery and multiple lanes in breaching a minefield. Specifically, this thesis develops KHAFJI, a high resolution combat simulation, and then illustrates the type of analysis possible through an example of the model's use.

D. KHAFJI

KHAFJI, employing user input parameters that define the minefield, the unit crossing the minefield, and the unit's tactical deployment, generates output which enables the user to compare various tactical options available to a maneuver commander in crossing the minefield. KHAFJI is a high resolution combat simulation written in SIMSCRIPT II.5 with SIMGRAPHICS I (detailed in Chapter III). By use of menu driven input screens, KHAFJI allows the user to quickly and easily define minefield scenarios. The user has a choice of multiple crossing lanes, size of crossing force, distribution of forces upon crossing lanes, multiple mine belts, and use of indirect fires against the minefield.

KHAFJI maintains a complete audit trail of significant events affecting the crossing unit. KHAFJI records the location of each mine, destruction of any mine by artillery, encounter of a mine by a member of the crossing unit, disablement of any member of

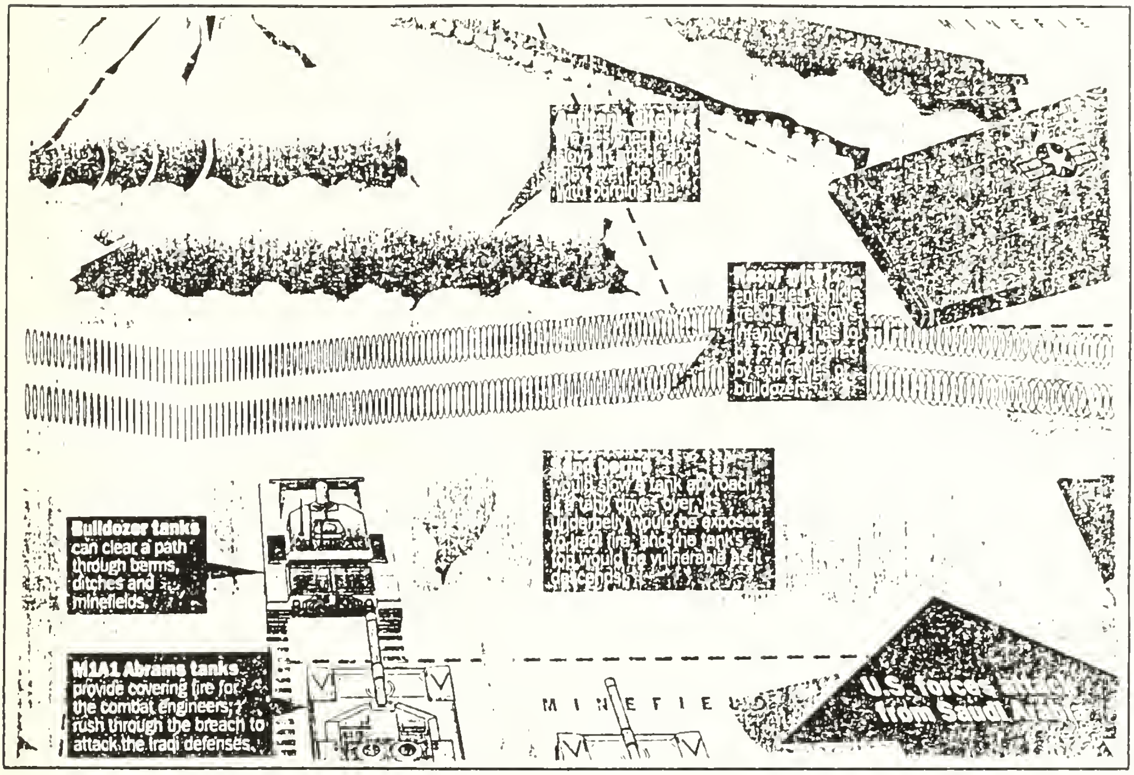

Figure 1. Iraqi Defensive Barrier

the crossing unit by any means, final status of each member of the crossing unit, total rounds of artillery fired, total number of unit members successfully transiting the minefield, and elapsed battle time (unit crossing time) (see Appendix C, Sample Output).

The data collected by KHAFJI can be used to analyze the relative effectiveness of various tactical options, for example are there more crossing unit members surviving the minefield using three crossing lanes with 80 rounds of artillery per lane than are unit members surviving using one lane to cross the minefield with no artillery support?

1. Graphics

A significant effort was expended in development of detailed graphics which display results as they occur during the simulation.

Using SIMGRAPHICS I software, KHAFJI displays the minefield and the unit as it, crosses the minefield. KHAFJI depicts each mine, each member of the crossing unit (animated as they cross the minefield), and each impacting artillery round. The graphics

provided by KHAFJI allows the user to see the crossing as it unfolds, thereby, reinforcing user confidence in the resultant data. (see Figure 2 on page 4).

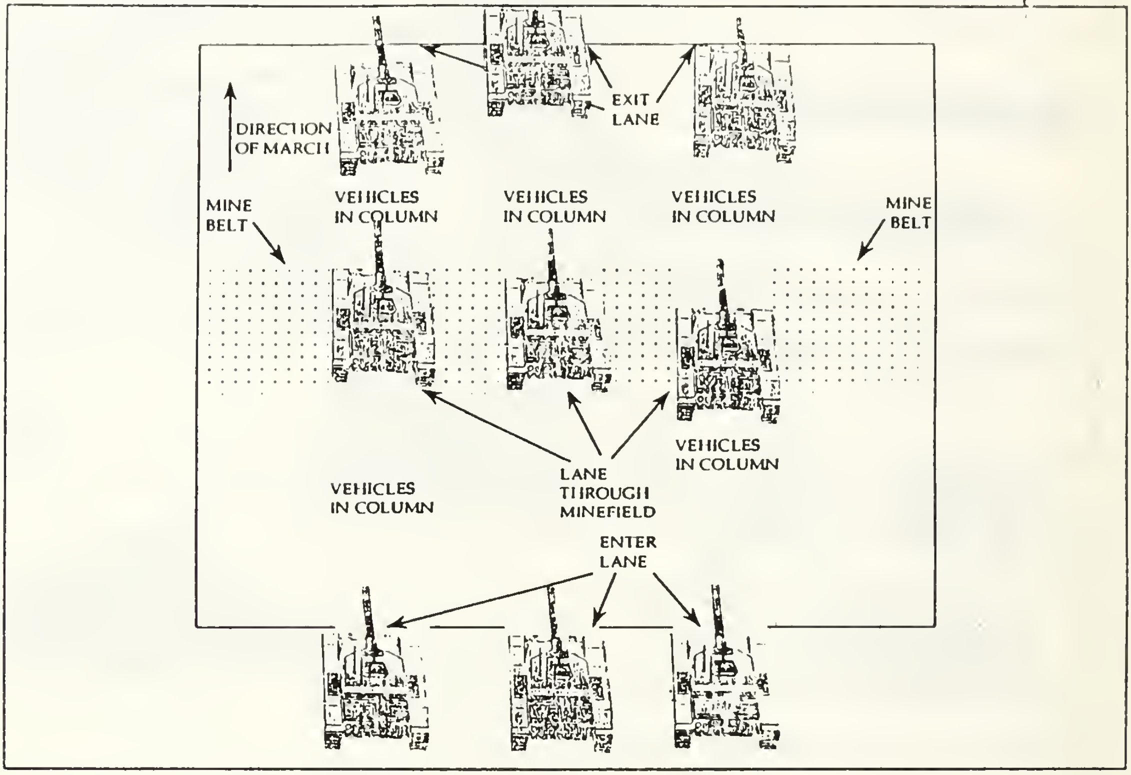

Figure 2. KHAFJI Display

2. Collaboration

KHAFJI was developed in association with Anderson, [Ref. 6]. The two simulations were developed sharing basic constructs for movement algorithms, to include detailed bypass algorithms. The simulations differ in that Anderson's thesis concentrates on modeling the tactics of using mine plows to clear minefields. This thesis models the tactics of using artillery fires to clear a minefield. Clearly, a useful follow on effort would combine the two models.

II. BACKGROUND

A. CURRENT COMBAT MODELS

Many combat models exist which contain minefield algorithms in the form of time delays or rolled-up casualty assessments such as:

JANUS - An interactive, stochastic, event driven, battalion/brigade model with individual element level resolution.

CASTFOREM - A stochastic, event driven, battalion/brigade model with individual element level resolution.

VIC - A deterministic, time step division level model with battalion level resolution.

However, these models are expensive and time consuming to maintain; require a degree of expertise to operate; and due to their complexity, sensitivity analysis of minefield specific concerns is difficult if not impossible to conduct. Furthermore, the above simulations do not model minefield effects at a sufficient level of resolution needed to investigate minefield breaching techniques. Therefore a high resolution model designed to address minefield specific concerns (KHAFJI) was required.

B. CURRENT DOCTRINE

Current U.S. Army doctrine, Airland Battle-Future, is offensive in nature and dependent upon maneuver. "Divisions will remain dispersed, preparing for battle...at the appropriate time, divisions will be committed and will maneuver to engage and defeat enemy forces...Emphasizing the importance of maneuver, we seek to avoid head-to-head attrition warfare...." Freedom of maneuver and maintenance of mobility is paramount. Therefore, obstacles, especially minefields, create an impediment to successful execution of AirLand Battle-Future. [Ref. 7]

C. TACTICAL OPTIONS

The unit of a maneuver force that would actually encounter and subsequently breach a minefield would be a company or platoon sized element (3 to 21 vehicles). Upon encountering a minefield obstacle, the maneuvering unit commander has four basic options available: bypass the minefield; plan, organize, and coordinate a deliberate breach; hastily breech the minefield; or force through the minefield. [Ref. 8]

1. Bypass

If possible, the maneuver commander should bypass any minefields encountered, reporting the minefield's location and maintaining the initiative. However this is frequently not a likely choice.

2. Forcing Through

Forcing-through the minefield (bulling through--simply driving through the minefield without any special preparations) is executed only when no other options exist. Any commander choosing this option would expect heavy casualties.

3. Deliberate Breach

A deliberate breach is conducted by engineer assets and adversely affects maneuver in terms of time and effort. To breach a minefield, a maneuver unit must:

- Suppress enemy weapons
- Obscure the enemy's observation
- Secure the far side of the minefield
- Reduce the minefield by clearing a path through.

4. Hasty Breach

A hasty breach (breaching the minefield quickly with little preparation) is accomplished with the unit commander's own assets, man-portable mine detectors, personnel, and indirect fire support.

D. MODEL DESIGN

1. Tactical Considerations

While traversing a minefield, the breaching unit is vulnerable to landmines and enemy covering fires. The faster the unit moves across the minefield, the less vulnerable the unit is to attrition by enemy covering fire (less exposure time), however, the unit's vulnerability to landmines is increased due to less time available to clear or avoid the mines. This produces a trade-off in speed to decrease enemy covering fire inflicted casualties versus increased landmine inflicted casualties.

The addition of indirect fires further complicates the tactical situation. Theoretically, using artillery fires against the minefield should help the crossing unit by reducing the density of the minefield thereby decreasing the frequency of mine encounter, however, damage of the terrain by indirect fires (cratering) will probably impede the unit's speed thereby increasing the unit's vulnerability to covering fires (increased exposure time).

The use of multiple lanes has the promise of decreasing the unit's crossing time. With the benefit of two or more lanes to cross the minefield, the unit should adopt shorter march columns, crossing simultaneously, thus realizing significantly shorter crossing times. The decrease in crossing time should decrease the unit's vulnerability to covering fires, however, the use of multiple lanes causes the unit to adopt a multi-vehicular front while crossing the minefield instead of the standard single vehicle front. This tactical posture increases the likelihood and rate of mine encounter, especially when bypassing a disabled vehicle (see Figure 3 on page 8).

Bypassing a disabled vehicle further complicates the scenario. If a vehicle becomes disabled, follow on vehicles must go around. This maneuver increases crossing time (each vehicle in column must use some lateral movement to bypass the disabled vehicle) and increases likelihood of mine encounter (vehicles must now traverse more of the breath and depth of minefield)..

2. Nature of KHAFJI

KHAFJI successfully models the tactical considerations detailed above to evaluate the hasty breach and force-through options.

KHAFJI allows the analyst to use combinations of indirect fire support and multiple lanes among many other user defined parameters (detailed in Chapter III) to determine the most efficient tactical option in terms of friendly survivors and time to cross minefield. KHAFJI helps answer the following questions:

- Should the unit commander simply force through the minefield?
- Is indirect fire support effective in reducing the minefield?
- Should the force use two or more lanes in the breaching the minefield?
- Is a combination of these options more effective due to synergism?

A high resolution model is needed to provide data at this level of detail. KHAFJI models a company-team/platoon sized armored force attacking through an enemy minefield. The minefield is covered by hostile direct fires. The attacking force is attrited by enemy direct fire, and landmines. The enemy force is played notionally, therefore, the enemy force is not attrited. The attacking force must use tactics as opposed to technology to overcome the obstacle. The friendly tactical arsenal includes the use of artillery against the minefield, a choice of multiple breaching lanes, and distribution of forces on the lane or lanes chosen.

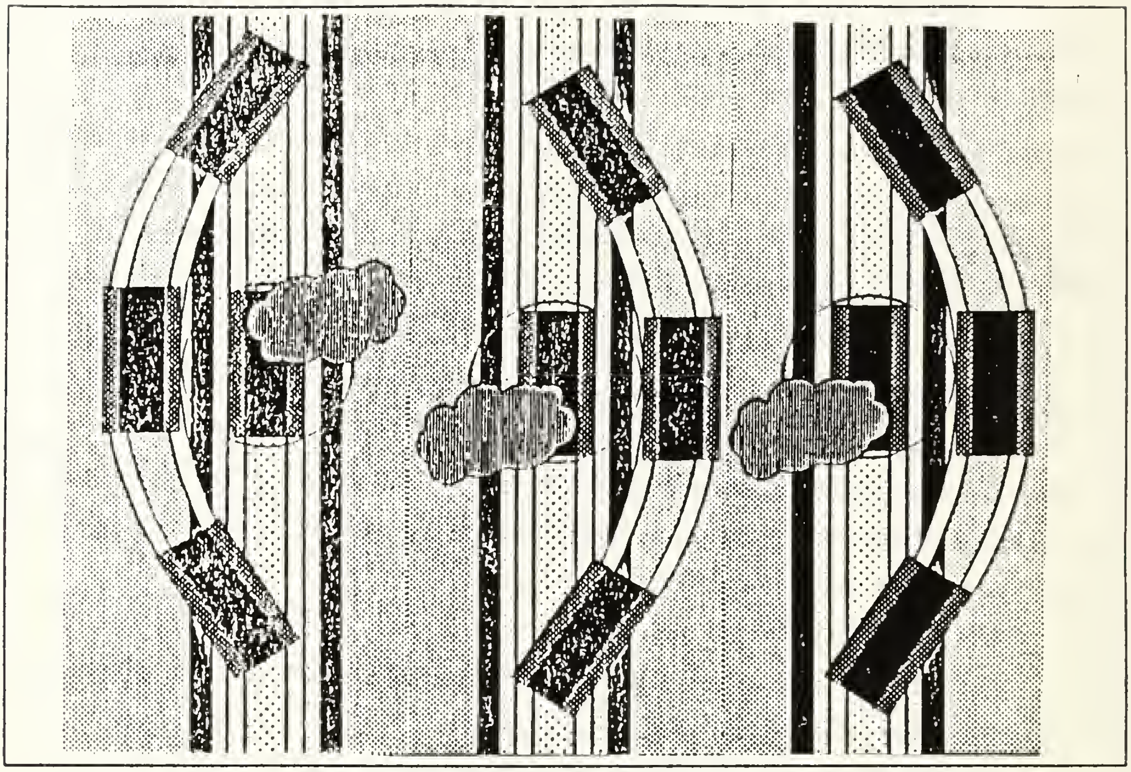

Figure 3. Multiple Lanes - Bypass of Disabled Vehicle

III. METHODOLOGY

KHAFJI is a high resolution combat simulation written in SIMSCRIPT II.5, with SIMGRAPHICS (graphics can be turned off if desired). The simulation is constructed around three permanent and two temporary entities. Permanent entities are SIMSCRIPT constructs used to represent passive elements in a simulation model. Permanent entities in KHAFJI are mine belts, lanes, and indirect fire weapons. Temporary entities are SIMSCRIPT constructs used to model objects which have a limited life in a simulation or vary in number during the simulation. Temporary entities in KHAFJI are mines and vehicles or combat elements. [Ref. 9]

Algorithm development followed a four phased approach:

- Build the Data Base
- Create the Minefield
- Fire Indirect Fire Weapons
- Move Through the Minefield.

A. BUILD THE DATA BASE

Data needed for each run is entered by the user through a series of menu driven screens.

1. General Data

This screen allows the user to design the minefield scenario. The user designates location, width, and length of the minefield, number of mine belts located within the minefield, number of lanes to use for the crossing, size of the breaching force, and the number of indirect fire weapons available (total of blue and red weapons) (see Table 2 on page 11).

a. Vehicle Dimensions

This screen allows the user to describe the breaching element. The user designates the breaching element's length, width, and speed among other parameters. The types of vehicles or breachers possible are limited only by the imagination of the user. Breacher types are currently homogeneous throughout the breaching force (see Table 3 on page 12).

2. Probability of Kill For Mines

This screen allows the user to assign probabilities of kill for mines ($P_k mine$) against the three breacher types designated in Vehicle Dimensions above (see Table 1 on page 10).

a. *Mine Belt Data*

This screen allows the user to tailor each mine belt to his specific scenario. The user defines the length and depth of the mine belt, along with the number of mines and type of mine located within the mine belt. Though mine type is homogeneous within mine belts, the user can establish a mixture of mine types within any belt by overlaying mine belt locations (see Table 4 on page 12).

3. Weapons Data

This screen allows the user to define each indirect fire weapon. The user establishes the weapon's unit, type, lethal radius, impact errors (northing and easting), and the number of volleys to fire. All weapons are modeled with a circular destructive region. (see Table 5 on page 13).

4. Lane Data

This screen allows the user to designate a starting point (lane entry point) and to distribute the crossing forces on selected lanes. Total forces distributed on the lanes must equal number of elements designated in General Data above (see Table 6 on page 13).

5. Iterative Control

This screen allows up to 1,000 iterations, each iteration containing identical "start-up" data as outlined above. If using more than four volleys of artillery per lane, graphics must be turned off depending upon storage capacity of host personal computer (see Table 7 on page 14).

Table 1. INPUT SCREEN 3

	DEFAULT	TYPE-1	TYPE-2	TYPE-3
MINE P_k	PRESSURE	0.01	0.02	0.80
	MAGNETIC	0.01	0.30	0.80
	CONTACT	0.50	0.30	0.20

B. CREATE THE MINEFIELD

Each mine is positioned to a tenth of a meter within a specified mine belt using a two-dimensional cartesian coordinate system, (x,y), which represents easting and north-

Table 2. INPUT SCREEN 1

	INPUTS	RANGE	DEFAULT	REMARKS
GENERAL DATA	GRID ZONE	CHARACTER	XX	
	EASTING	1 - 99	56	
	NORTHING	"	78	
	WHETHER TO FIRE INDIRECT WEAPONS AGAINST MINE FIELD	CHARACTER	YES	
	# MINE BELTS	1 - 4	1	
	WIDTH OF MINE FIELD	1 - 1000	200	
	RADIUS OF MINE	0.1 - 5	0.2	
	# LANES	1 - 9	1	
	# BREACHERS	1 - 50	6	
	DISTANCE TO TRAVEL	25 - 1000	100	
	# NUMBER OF INDIRECT WEAPONS	1 - 9	2	

ing, respectively, of the military standard grid reference system. In an effort to insure homogeneity of mine placement throughout the width and depth of the minefield, while avoiding overlapping of mine locations, the distribution of mine placement is modeled as bivariate normal. This formulation provides a realistic mine location error around a central point determined for each individual mine i.e., $(X, Y) \sim BVN(\mu_x, \mu_y, \sigma_x, \sigma_y)$ (see Figure 4 on page 15). Mine types (i.e., 1-pressure, 2-magnetic, and 3-contact) are homogeneous within each belt.

Table 3. INPUT SCREEN 2

	INPUTS	RANGE	DEFAULT	REMARKS
VEHICLE DIMEN-SIONS	LENGTH	1 - 20	9.83	METERS (M1A1)
	WIDTH	"	3.48	"
	TRACK WIDTH	"	0.63	"
	TYPE	1	1	1-TANK (M1A1)
	RATE KILLED	0 - 10	0.03	KILLS PER MIN
	KILL RATE STANDARD DEVIATION	0 - 10	0.09	"
	SPEED	0.1 - 500	85	METERS PER MIN

Table 4. INPUT SCREEN 4

	INPUTS	RANGE	DEFAULT	REMARKS
MINE BELT DATA	DEPTH OF BELT	0 - 2000	50	METERS INTO MINE FIELD
	LENGTH OF MINE BELT	1 - 200	50	"
	# MINES IN BELT	1 - 1000	10	
	MINE TYPE	1,2,3	1	1-PRESSURE 2-MAGNETIC 3-CONTACT

1. Easting

Mine location easting is determined by computing uniform intervals across the width of the minefield, then determining a standard deviation and mean for each interval.

(1) Interval . The interval computation segments the minefield width into a discrete number of cells where:

Table 5. INPUT SCREEN 5

	INPUTS	RANGE	DEFAULT	REMARKS
WEAPONS DATA	WEAPON TYPE	CHARACTER	HOWITZER	
	WEAPON UNIT	"	BLUE	BLUE OR RED
	WEAPON LETHAL RADIUS	0.1 - 50	10	METERS
	IMPACT EASTING ERROR	"	"	
	IMPACT NORTHING ERROR	"	"	
	# VOLLEYS TO FIRE	1 - 20	4	PER WEAPON

Table 6. INPUT SCREEN 6

	INPUTS	RANGE	DEFAULT	REMARKS
LANE DATA	# BREACHERS THIS LANE	1 - 50	6	SUM TOTAL MUST EQUAL TOTAL # BREACHERS
	LANE EASTING	0 - 999	561	LANE ENTRY POINT

$$INTERVAL = \frac{\text{MINE FIELD WIDTH}}{\text{\#MINES}}$$

(2) Mean . The mean computation defines the center of each individual cell as the distributional mean where:

Table 7. INPUT SCREEN 7

	INPUTS	RANGE	DEFAULT	REMARKS
ITERATIVE CONTROL	DISPLAY GRAPHICS	YES OR NO	YES	IF MORE THAN 4 VOLLEYS CHOSEN TURN OFF
	# ITER-ATIONS	1 - 1000	5	# OF RUNS

$$\mu_x = \frac{(i \times 2 - 1) \times INTERVAL}{2}, \text{ where } i = 1,..., \#MINES.$$

See Figure 4 on page 15.

(3) Standard Deviation. The standard deviation computation prevents overlapping of eastings by restricting each cell distribution within its target cell. The standard deviation is approximated by the interval length divided by six since 99% of the area under a normal curve is ± 3 standard deviations from the mean.

$$\sigma_x = \frac{INTERVAL}{6}$$

The easting, x, is then computed by sampling from a normal distribution, $X \sim N(\mu_x, \sigma_x)$.

2. Northing

Mine location northing is determined by random sampling from a normal distribution using user provided northing of the mine belt as a mean and user input mine belt length divided by six as the standard deviation, $Y \sim N(\mu_y, \sigma_y)$, where:

$$\mu_y = \text{MINE BELT NORTHING}$$

$$\sigma_y = \frac{\text{MINE BELT LENGTH}}{6}$$

This standard deviation computation, along with designating the mine belt northing as the center of the mine belt allows mine northings to be normally distributed throughout the length of the mine belt.

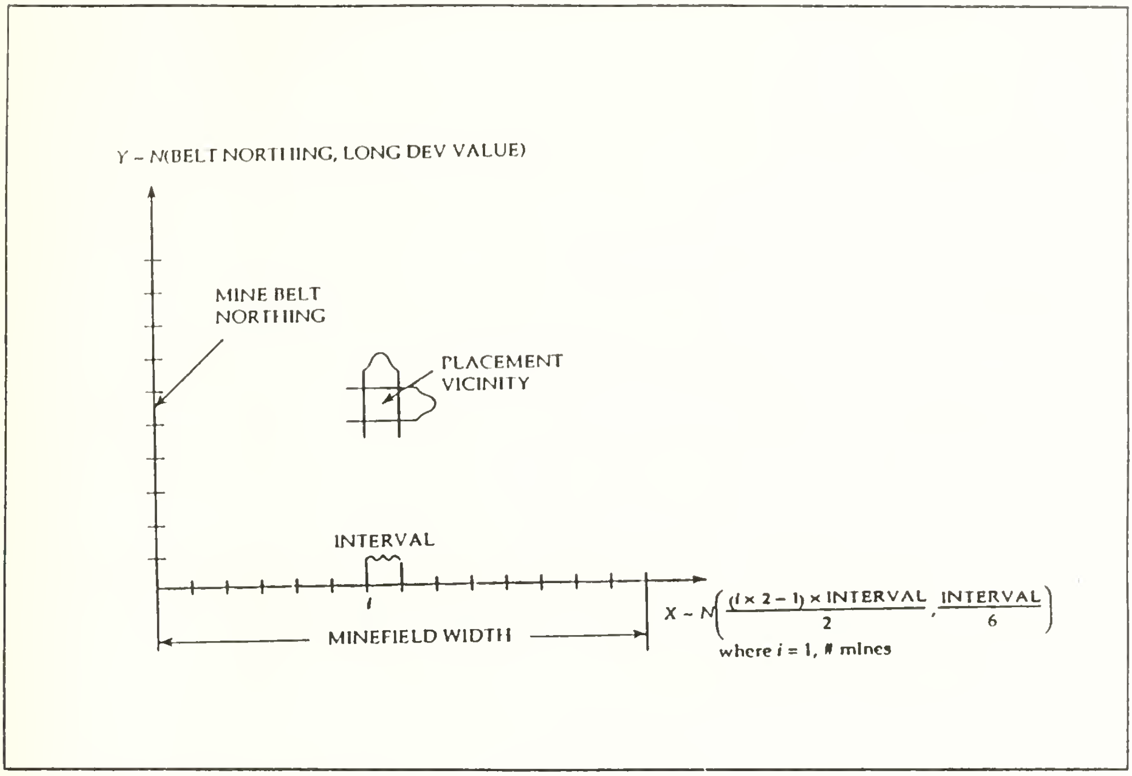

Figure 4. Mine Placement

C. INDIRECT FIRE WEAPONS

Indirect fire weapons are modeled as cookie cutter weapons, [Ref. 10], i.e, firing errors are circular normal and lethal radius is a constant (user provided). Currently artillery fires are used against the minefield only.

1. Targeting

The key concern of artillery targeting is target coverage. For the targeting of a lane through a minefield, a methodology which ensures coverage of the entire length of the minefield is required (crossing unit does not know location of mine belt). Therefore, a methodology based on the minefield length and the lethal radius of the particular weapon firing was chosen. This methodology provides consistent and homogeneous coverage of the minefield regardless of the type of weapon firing. Therefore, the number and location of indirect fire weapon aimpoints are determined internally as functions of the minefield lane entry point, the user provided distance the crossing unit intends to

travel, and the lethal radius of each indirect fire weapon as follows (see Figure 5 on page 17):

$$\# \text{ AIMPOINTS} = \frac{(DISTANCE)}{(2 \times \text{LETHAL RADIUS})}$$

$$\text{AIMPOINT NORTHING} = (i \times 2 - 1) \times \text{LETHAL RADIUS}, \text{ where } i = 1, \ldots, \# \text{ AIMPOINTS}$$

$$\text{AIMPOINT EASTING} = \text{LANE ENTRY EASTING}$$

EXAMPLE:

DISTANCE TO TRAVEL = 100

LETHAL RADIUS = 10

LANE ENTRANCE = 5610 7800 (eight place grid coordinate, where 5610 is the easting, and 7800 is the northing. The last digit in easting and northing represents tens of meters.)

$$\text{NUMBER OF AIMPOINTS} = \frac{100}{2 \times 10} = 5$$

AIMPOINTS:

5610 7801

5610 7803

5610 7805

5610 7807

5610 7809.

2. Impact locations

Keeping with the generally accepted convention that artillery impacts are normally distributed, impact locations are modeled as bivariate normal, using aimpoint ordinates as means and the weapon's impact errors as standard deviations. [Ref. 10].

$$\text{IMPACT EASTING} \sim N(\text{AIMPOINT EASTING, ERROR IN EASTING})$$

$$\text{IMPACT NORTHING} \sim N(\text{AIMPOINT NORTHING, ERROR IN NORTHING})$$

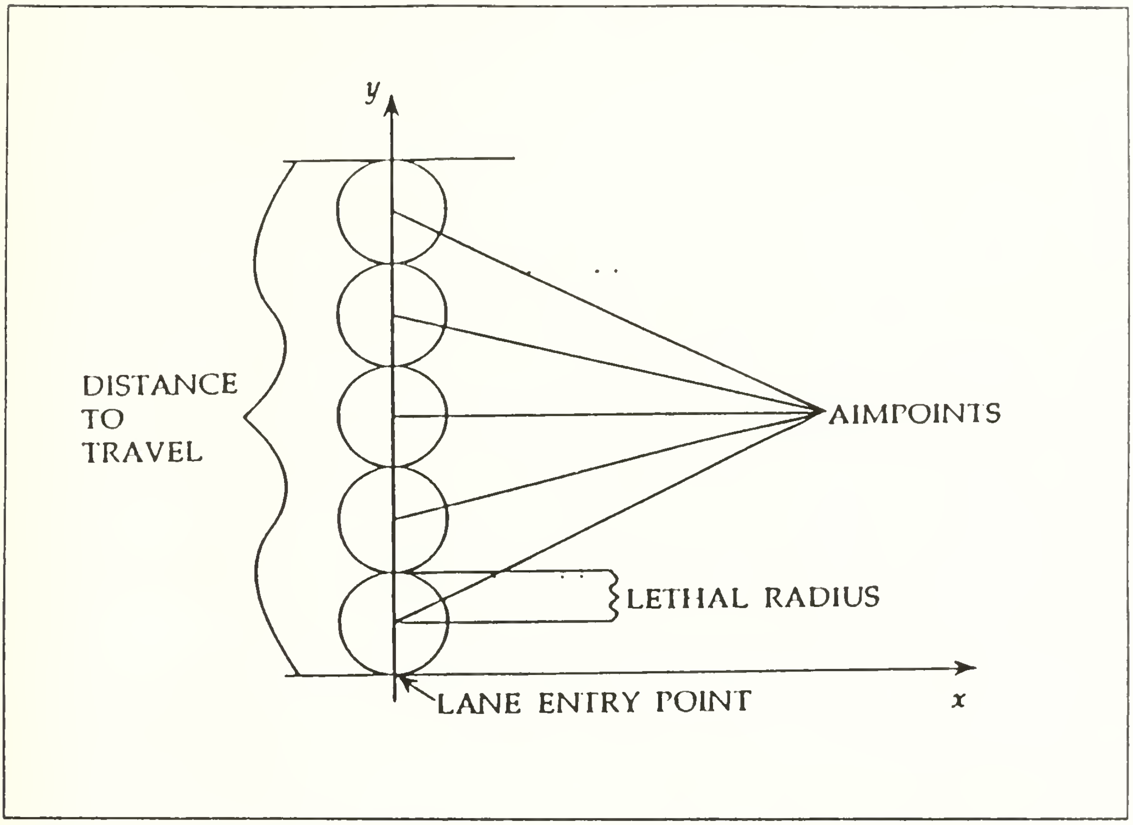

Figure 5. Indirect Fire Aimpoints

3. **Battle Damage Assessment**

The distance from munition impact to target is determined by geometry i.e., $DISTANCE = \sqrt{X^2 + Y^2}$, where X = MISS DIST IN EASTING and Y = MISS DIST IN NORTHING.

Destruction of a mine by indirect fire is achieved if mine is located within the weapon's lethal radius (see Figure 6 on page 18).

4. **Terrain**

To model the effects of indirect fire damage to the terrain, the velocity of each breacher traversing the minefield is degraded by a user input percentage for volleys to be fired into the minefield. For example using a 5% default we have:

INITIAL VELOCITY = 85 METERS PER MINUTE

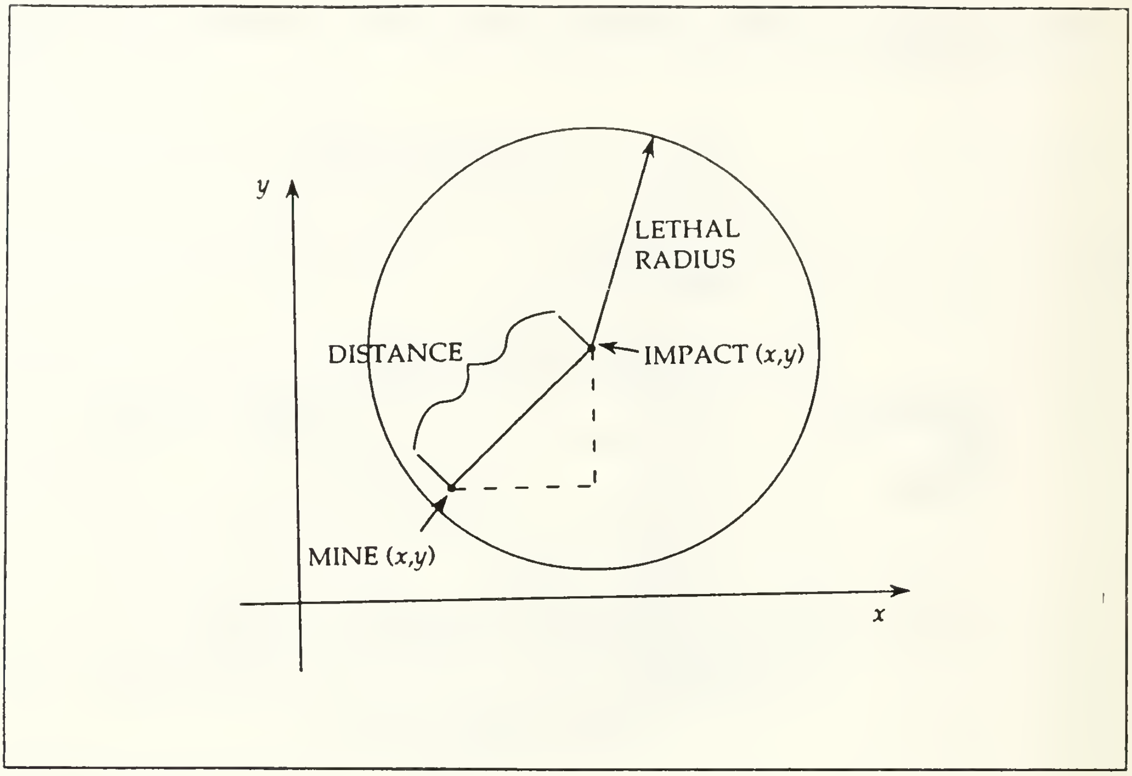

Figure 6. Battle Assessment

VOLLEYS	% DEGRADATION OF VELOCITY	RESULTANT VELOCITY
0	0%	85
1	5%	80.75
2	10%	76.5
3	15%	72.25
4	20%	68.

This degradation is a one time computation computed during the initialization phase of the simulation. The number of volleys to fire is a user input.

D. MOVEMENT

To approximate reality as closely as time and other resource permit, KHAFJI's movement algorithms model the actions of individual vehicles crossing the minefield. The vehicles are modeled in column formation with one vehicle following the column's lead vehicle along predetermined lanes. Within each lane, each vehicle is further given a computed path to follow. Each individual path is uniquely determined by random

normal sampling along a central axis bisecting the lane(s) through the minefield. This methodology is intended to model the variations in paths realized with one vehicle attempting to follow another (driving errors) (see Figure 7 on page 20).

As discussed in Chapter II (paragraph D.1.,Tactical Considerations), if a vehicle is disabled, following vehicles must bypass the disabled vehicle. KHAFJI using constructs developed by Anderson, [Ref. 6], models vehicles as circular objects. In bypassing a disabled vehicle, KHAFJI computes the area of the disabled vehicle, then determines a peripheral path (tangent to circle representing disabled vehicle) around the disabled vehicle (see Figure 8 on page 21 and Appendix D, paragraph T, Bypass).

Breaching elements will be in one of the four following states while negotiating the minefield (movement algorithm developed with Anderson), [Ref. 6]:

- Enter the Minefield
- Encounter Mine
- Kill
- Exit Minefield.

1. Enter Minefield

Modeling direct fire casualties presented somewhat of a dilemma. The methods available were reduced to two options: determine direct fire casualties as a result of individual direct fire engagements or determine direct fire casualties by use of Lanchester kill rate coefficients. Due to resource constraints the Lanchester option was selected.

All breachers will enter the minefield. Upon entering the minefield, each breacher is given a "time to death", a time in which the breacher will be rendered a direct fire casualty if he has not exited the minefield. This death time is modeled as normally distributed with a mean death time equal to the inverse of the rate at which red kills blue i.e.,red kill rate (RKR). To lend variation to the death times, a death time standard deviation was developed (σ_{DT}). No historical data were available to gauge the dispersion of direct fire casualty times realized by a unit crossing a minefield, therefore, σ_{DT} is a user input parameter. A default value equal to the inverse of RKR divided by three is provided. This default value provides a normal curve, distributed over an interval from zero to twice the mean value.

$$RKR = \frac{KILL}{TIME} \quad \text{(KILL PER TIME)}$$

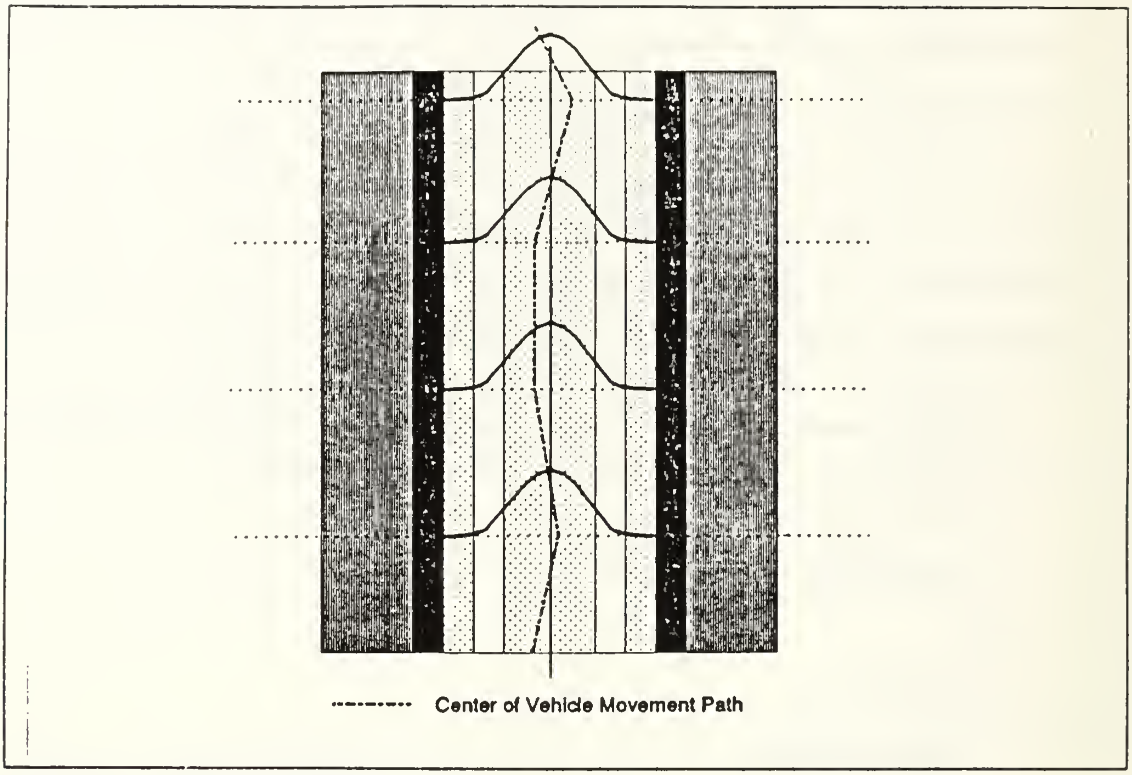

Figure 7. Minefield Path with Driving Errors

$$\frac{1}{RKR} = \frac{TIME}{KILL} = \text{MEAN DEATH TIME (TIME PER KILL)}$$

$$\sigma_{DT} = \frac{1}{(3 \times RKR)}$$

$$\text{DEATH TIME} \sim N\left(\frac{1}{\text{RED KILL RATE}}, \sigma_{DT}\right).$$

Individual element death times are determined by random sampling from the above normal distribution.

2. Encounter Mine

Once a mine is encountered by a breacher, element kill is determined by Monte-Carlo, Uniform (0,1) random sampling against the user input mine P_k. Whether or not a breacher is killed, the encountered mine is rendered inactive.

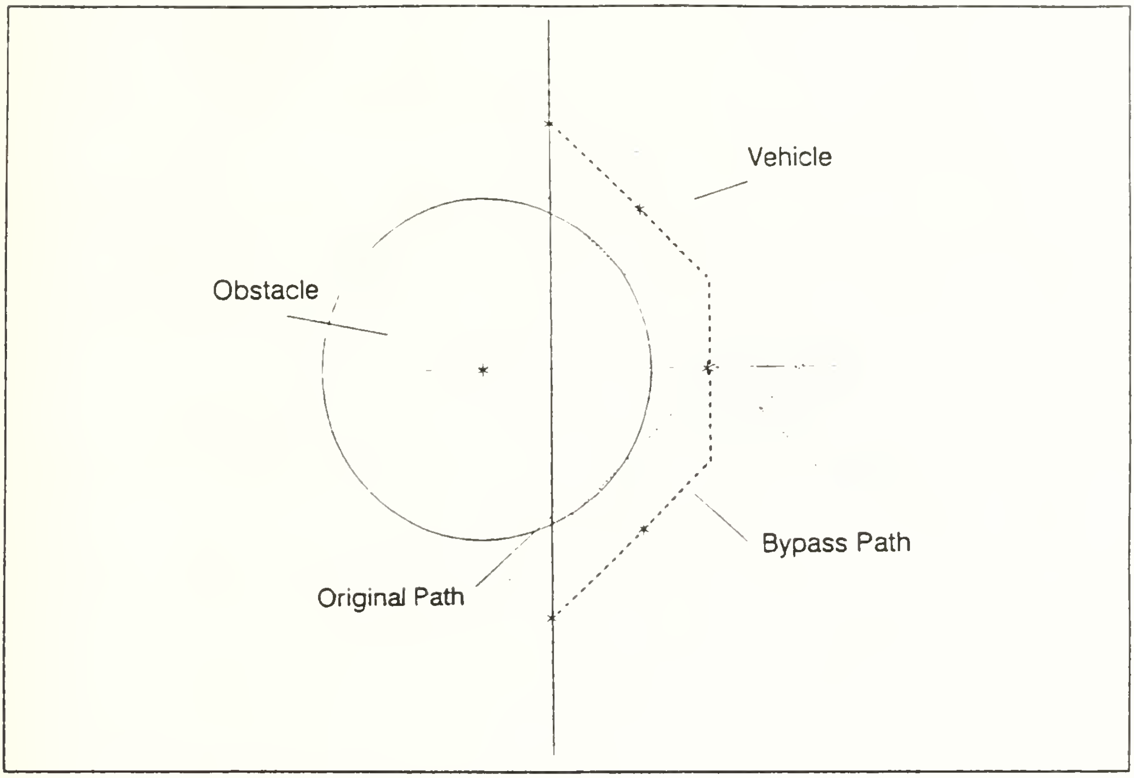

Figure 8. Bypass Geometry

3. Kill

A breacher enters a state of kill by mine encounter or enemy direct fire--time in minefield exceeds death time.

4. Exit Minefield

If the breaching element negotiates the minefield within its "death time", it successfully exits the minefield. The simulation terminates when all breachers have either successfully transited the minefield or have become casualties.

IV. ANALYSIS

A. EXAMPLE ANALYSIS

The intent of this chapter is to provide an example of the kind of analysis that can be accomplished with KHAFJI. This example analyzes the effectiveness of using multiple lanes and field artillery fires to support a unit crossing a minefield obstacle.

For a unit crossing a minefield the most important aspects are crossing the minefield with minimal casualties and crossing the minefield in the shortest length of time. This time factor is important because an attacking unit cannot afford to loose its offensive momentum. Furthermore, the longer a unit stays in a minefield the greater the likelihood of its members becoming casualties.

KHAFJI output includes such measures of effectiveness (MOEs) as SURVIVORS (total unit members successfully exiting the minefield) and BATTLE TIME (unit's elapsed crossing time). Additionally, a complete audit trail of significant events is provided (see Appendix C, Sample Output). These MOEs and accompanying data will be used to evaluate several tactical options. Additionally, sensitivity analysis will be conducted on the volume of artillery fired into the minefield.

B. SETTING

Data were collected from KHAFJI runs to evaluate the MOEs of SURVIVORS and BATTLE TIME, in scenarios involving multiple lanes without friendly artillery fire support and with varying levels of friendly artillery.

1. Scenarios

Five replications of each of fifteen scenarios were conducted as follows:

- 1 Lane Without Artillery Support
- 2 Lanes Without Artillery Support
- 3 Lanes Without Artillery Support
- 1 Lane With Artillery Support (20,40,60,80 ROUNDS PER LANE)
- 2 Lanes With Artillery Support " "
- 3 Lanes With Artillery Support " "

See Table 8 on page 23.

Table 8. RUN DESIGN MATRIX

MOEs	WITH INDIRECT FIRE (80,60,40,20) (ROUNDS PER LANE)				W/O INDIRECT FIRE
1 LANE	X_{11}	X_{12}	X_{13}	X_{14}	X_{15} (*BASECASE*)
2 LANES	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}
3 LANES	X_{31}	X_{32}	X_{33}	X_{34}	X_{35}

2. Run Parameters

- Sample Size = 5
- Minefield Density = 0.25 mines per meter
- P_k mine = 1.0
- # rounds = 0, 20, 40, 60, 80 per lane
- # breachers = 7 (total number of breachers distributed on lanes)
- Distance to travel = 100 meters
- Speed = 85 meters per minute
- Mean Death Time = 33 minutes.

C. DATA ANALYSIS

The data were analyzed using analysis of variance (ANOVA) for parametric analysis and Kruskal-Wallis non-parametric test for equal location parameters.

Analysis is further supported by Bartlett's test of homogeneity of variance. Bartlett's test tests homogeneity of variances of populations assumed to be normally distributed (test statistic is a ratio of the weighted geometric mean of the variances to the weighted arithmetic mean of the variances, where weights are relative degrees of freedom). A significant p-value indicates either non-normality or inequality of distribution variances (see Appendix A, Data Analysis). [Ref. 11]

The data is also analyzed graphically using box plots and bar graphs. A box plot is a compact graphical method of displaying data distributions. The box covers the distributions interquartile range, observation falling within the 25^{th} to 75^{th} quartiles, with extending limbs depict observations occurring outside the interquartile range. The length of the box gives a relative measure (in comparison to companion plots) of distributional variance (see Figure 9 on page 26).

For MOE, SURVIVORS, for each of the lane options (1 lane, 2 lanes, 3 lanes) the null hypothesis tested is: all run means are equal versus the alternate hypothesis that mean number of survivors with indirect fire support are greater than mean number of survivors without indirect fire support:

H_0: $\mu_0 = \mu_{20} = \mu_{40} = \mu_{60} = \mu_{80}$, where 0,20,40,60,80 indicate rounds of artillery per lane.

H_A: At least two run means differ.

A significance level of $\alpha = 0.05$ was used.

For MOE, BATTLE TIME, for each of the lane options (1 lane, 2 lanes, 3 lanes) the null hypothesis tested is all run means are equal versus the alternate hypothesis that mean battle time with indirect fire support is less than mean battle time without indirect fire support:

H_0: $\mu_0 = \mu_{20} = \mu_{40} = \mu_{60} = \mu_{80}$, where 0,20,40,60,80 indicate rounds of artillery per lane.

H_A: At least two run means differ.

A significance level of $\alpha = 0.05$ was used.

D. RESULTS

To confirm data suitability for parametric analysis of variance the Bartlett test for homogeneity of variance, using a significance level of $\alpha = 0.05$, was utilized.

1. Survivors

a. 1 Lane

The Bartlett test yields a significance level of 0.04, therefore we conclude the data to be suitable for parametric ANOVA. Testing the null hypothesis that the five run distributions of survivors (1 lane with 0, 20, 40, 60, and 80 rounds of artillery) are equally distributed versus the alternate hypothesis that at least two distributions differ, yields levels of significance of 0.36 and 0.20 for parametric and non-parametric ANOVA, respectively, therefore we fail to reject the null hypothesis at $\alpha = .05$ level of significance. We conclude that with the single lane option there is no significant difference in mean survivors (see Appendix A, paragraph A.1).

b. 2 Lanes

The Bartlett test yields a significance level of 0.49, therefore we conclude the data to be unsuitable for parametric ANOVA. Using Kruskal-Wallis only to test the

null hypothesis that the five run distributions of survivors (2 lanes with 0, 20, 40, 60, and 80 rounds of artillery per lane) are equally distributed versus the alternate hypothesis that at least two distributions differ, yields a significance level of 0.0009, therefore we reject the null hypothesis in favor of the alternate hypothesis at $\alpha = .05$ level of significance. We conclude that with the two lane option the number of survivors significantly increase as the level of artillery support per lane increases. (see Appendix A, paragraph A.2.).

c. 3 Lanes

The Bartlett test yields a significance level of 0.74, therefore we conclude the data to be unsuitable for parametric ANOVA. Using Kruskal-Wallis only to test the null hypothesis that the five run distributions of survivors (3 lanes with 0, 20, 40, 60, and 80 rounds of artillery per lane) are equally distributed versus the alternate hypothesis that at least two distributions differ, yields a significance level of 0.0009, therefore we reject the null hypothesis in favor of the alternate hypothesis at $\alpha = .05$ level of significance. We conclude that with the three lane option the number of survivors significantly increase as the level of artillery support per lane increases (see Appendix A, paragraph A.3.).

d. Conclusions

From parametric and non-parametric analysis we conclude that for two and three lane options, the number of survivors increase as the level of artillery support per lane increases. There is no significant difference in survivors within the one lane option.

From Figure 10 on page 27, which depicts mean SURVIVORS as a function of rounds of artillery fired per lane and number of lane(s) used, we see that for the single lane option, casualties are basically uniform across all levels of artillery support. For two and three lane options (see Figure 10 on page 27) we see a significant increase in mean SURVIVORS from 0 to 40 rounds of artillery per lane, leveling at 40 and 60 rounds, finally peaking at 80 rounds per lane. We further note that there is no significant difference in mean SURVIVORS between the two and three lane options.

Box plots presented (see Figure 9 on page 26 and Appendix A) further support the above conclusions.

A review of the run data (see Appendix B) reveals that segregating casualties into two groups, kills by mine and kills by direct fire, indicates another trend. Within single lane options, casualties are predominantly split evenly between mine and direct fire. However, within multiple lane cases, casualties are predominately mine kills. This

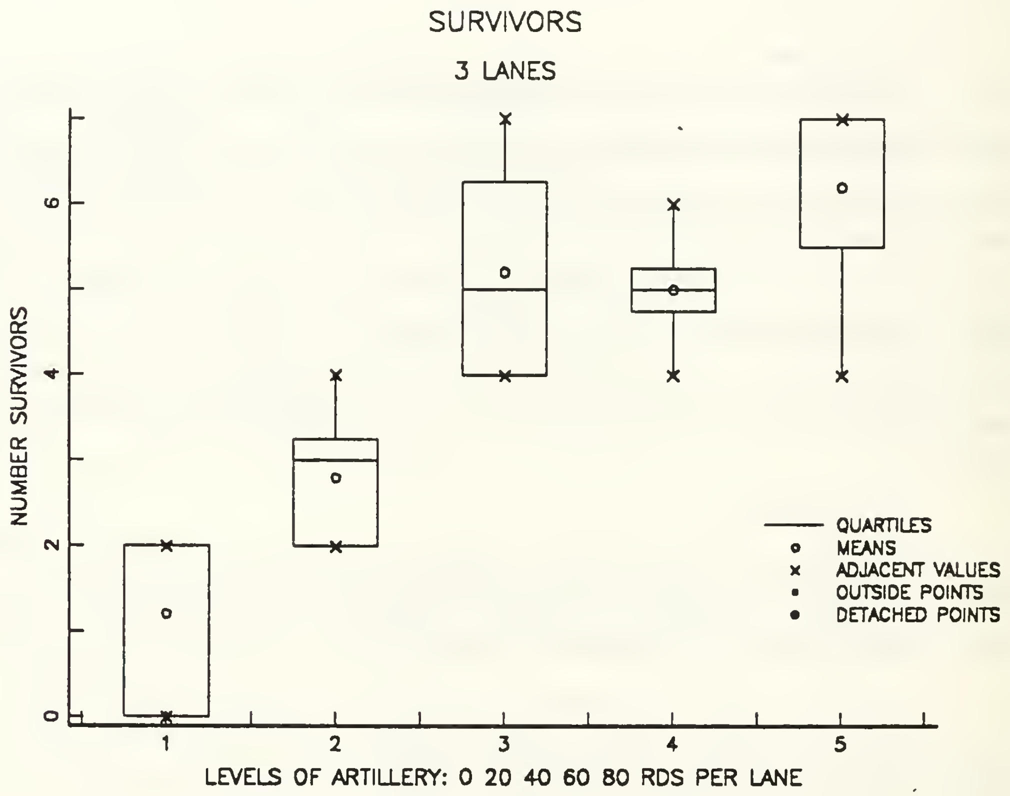

Figure 9. Box Plot-SURVIVORS-3 Lanes

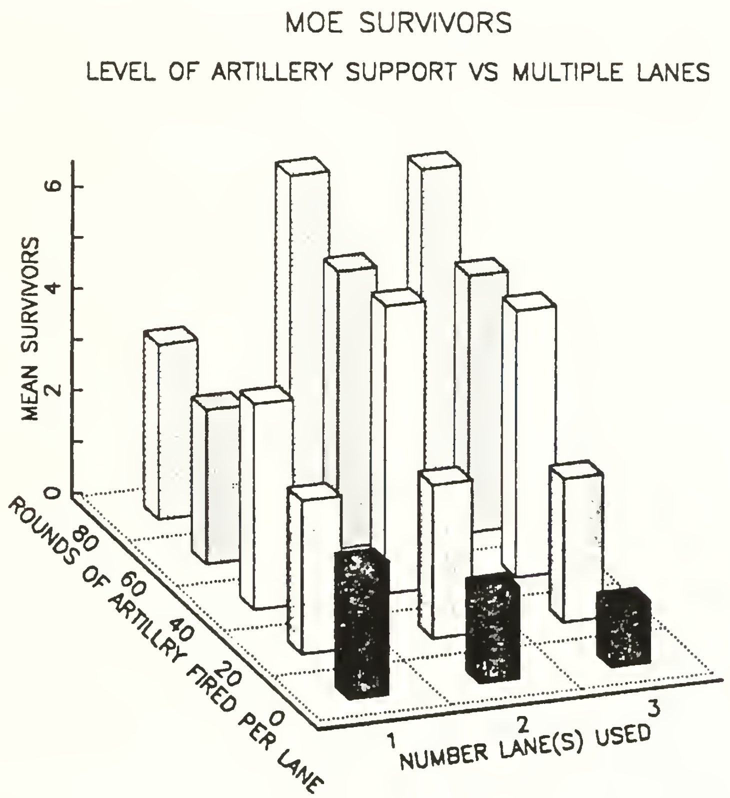

Figure 10. MOE-SURVIVORS

phenomenon suggests that factors within multiple lane cases lessen the effectiveness of direct fire while enhancing the opportunity for mine engagement (this trend is likely driven by the fact that $P_k$mine = 1.0). These factors are determined to be battle time (less time to employ direct fire) and order of march, becoming the lead vehicle (first to encounter the minebelt). This trend holds across all treatments with or without artillery support.

2. Battle Time

a. 1 Lane

The Bartlett test yields a significance level of 0.04, therefore we conclude the data to be suitable for parametric ANOVA. Testing the null hypothesis that the five run distributions of survivors (1 lane with 0, 20, 40, 60, and 80 rounds of artillery) are equally distributed versus the alternate hypothesis that at least two distributions differ, yields levels of significance of 0.0000004 and 0.01 for parametric and non-parametric ANOVA respectively, therefore, we reject the null hypothesis in favor of the alternate hypothesis at $\alpha = .05$ level of significance. We conclude that with the single lane option the mean battle time significantly increases as the level of artillery support per lane increases (see Appendix A, paragraph B.1.).

b. 2 Lanes

The Bartlett test yields a significance level of 0.01, therefore we conclude the data to be suitable for parametric ANOVA. Testing the null hypothesis that the five run distributions of survivors (2 lanes with 0, 20, 40, 60, and 80 rounds of artillery per lane per lane) are equally distributed versus the alternate hypothesis that at least two distributions differ, yields levels of significance of 0.000000002 and 0.0002 for parametric and non-parametric ANOVA respectively, therefore, we reject the null hypothesis in favor of the alternate hypothesis at $\alpha = .05$ level of significance. We conclude that with the two lane option the mean battle time significantly increases as the level of artillery support per lane increases (see Appendix A, paragraph B.2.).

c. 3 Lanes

The Bartlett test yields a significance level of 0.00002, therefore we conclude the data to be suitable for parametric ANOVA. Testing the null hypothesis that the five run distributions of survivors (3 lanes with 0, 20, 40, 60, and 80 rounds of artillery per lane) are equally distributed versus the alternate hypothesis that at least two distributions differ, yields levels of significance of 0.00000009 and 0.0003 for parametric and non-parametric ANOVA respectively, therefore, we reject the null hypothesis in favor of the alternate hypothesis at $\alpha = .05$ level of significance. We conclude that with the

three lane option the mean battle time significantly increases as the level of artillery support per lane increases (see Appendix A, paragraph B.3.).

d. Conclusions

From parametric and non-parametric analysis we conclude that across all lane options BATTLE TIME increases as the level of artillery support per lane increases.

Viewing Figure 11 on page 30, which depicts mean BATTLE TIME as a function of rounds of artillery fired per lane and number of lane(s) used, we see that mean BATTLE TIME significantly increases as the level of artillery support increases (speed is impeded due to terrain damage caused by impacting artillery--speed degradation factor discussed in Chapter III), and that mean BATTLE TIME decreases as the number of lanes used increase (breachers, in separate columns are able to cross simultaneously. Furthermore, viewing Figure 12 on page 31, we see that with the 3 lane option, using 80 rounds of artillery per lane, there is little or no variation in mean battle time.

Box plots presented (see Figure 12 on page 31 and Appendix A) further support the above conclusions.

E. COMPARISON OF OPTIONS

The "best" option under the given scenario is the option which yields maximum SURVIVORS at the minimum BATTLE TIME (see Table 9 on page 32).

In terms of SURVIVORS, KHAFJI yields a tie between two and three lanes, with 80 rounds per lane, there is no statistical difference between 6.4 and 6.2 (respective standard deviations yield overlapping 95% confidence intervals i.e., $\overline{X}_{ij} \pm 1.96\sigma$). Using BATTLE TIME as a final determinant, the best option is to cross with three lanes, using 80 rounds of artillery per lane (240 rounds total). This option allows maximum survivors in a minimum of crossing time. If conservation of artillery ammunition is more critical than time, then two lanes with 80 rounds of artillery per lane (160 rounds total) should be utilized.

F. OBSERVATIONS

- The use of multiple lanes when crossing a minefield obstacle decreases the crossing time of the unit, decreasing the unit's exposure time, thereby decreasing the frequency of attrition by direct fire. However, the use of multiple lanes increases the frequency of mine encounter.
- The use of artillery against the minefield in levels of 40 to 80 rounds per lane was significant in reducing the density of the minefield. However, use of artillery does significantly increase minefield transit time (it should be noted that this effect is sensitive to the user provided speed degradation factor).

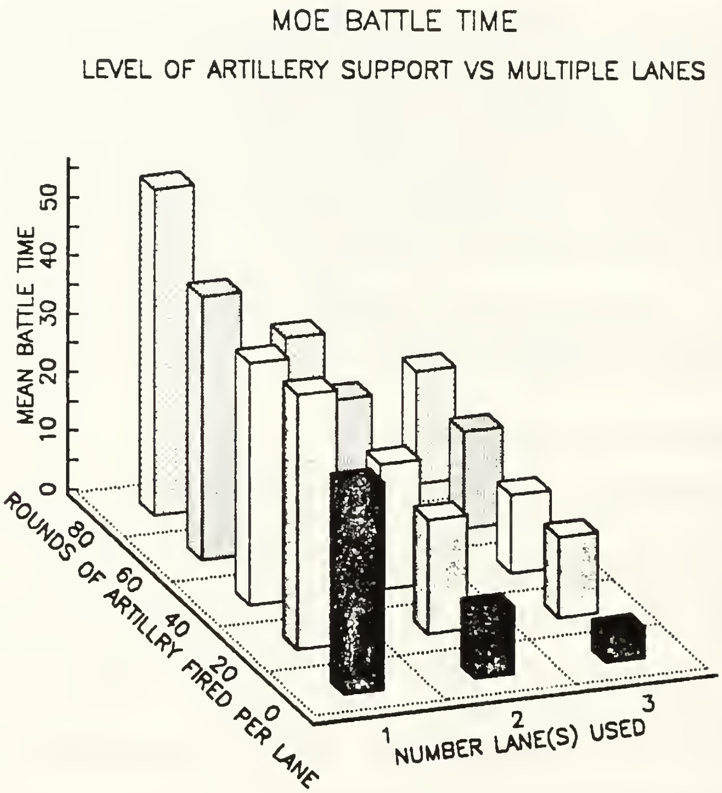

Figure 11. MOE-BATTLE TIME

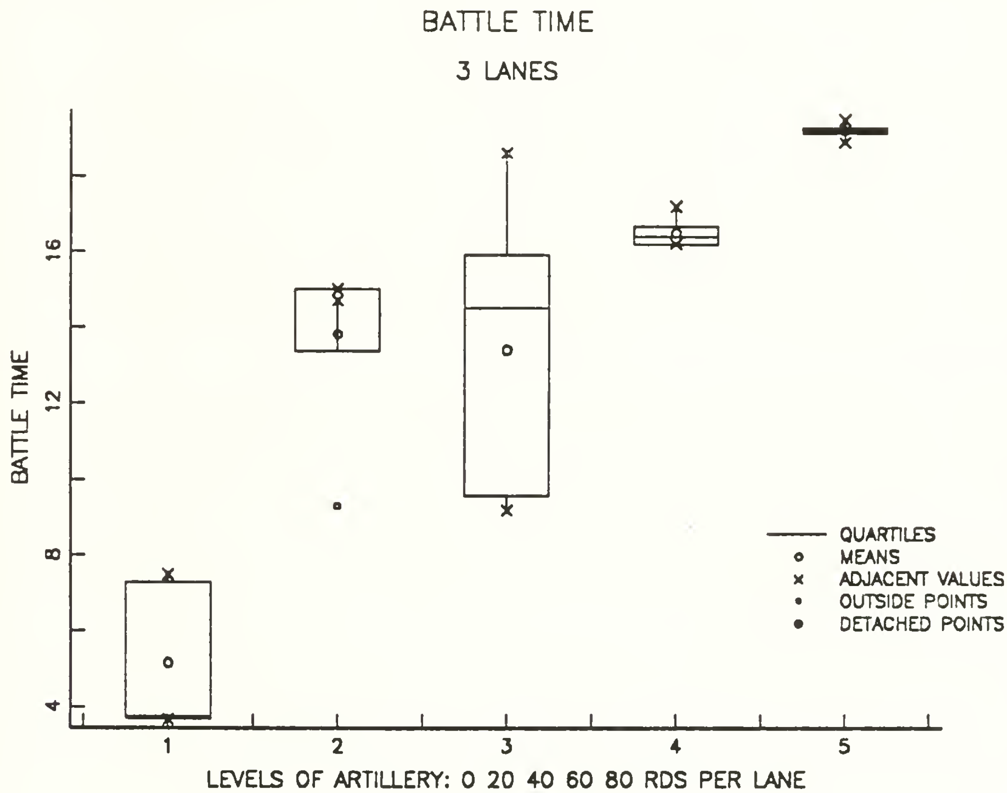

Figure 12. Box Plot-BATTLE TIME-3 Lanes

Table 9. MEAN MOE DATA

OPTIONS	SURVIVORS	BATTLE TIME
1 LANE	2.6	35.6
2 LANES	1.8	11.5
3 LANES	1.2	5.2
1 LANE (20 RDS ARTY)	3	43.6
2 LANES (20 RDS ARTY)	3	19.5
3 LANES (20 RDS ARTY)	2.8	13.8
1 LANE (40 RDS ARTY)	4	41.3
2 LANES (40 RDS ARTY)	5.6	21.4
3 LANES (40 RDS ARTY)	5.2	13.4
1 LANE (60 RDS ARTY)	3	45
2 LANES (60 RDS ARTY)	5.4	24.7
3 LANES (60 RDS ARTY)	5	16.5
1 LANE (80 RDS ARTY)	3.4	55.8
2 LANES (80 RDS ARTY)	6.4*	27.6
3 LANES (80 RDS ARTY)	6.2*	19.2*

G. CAUTIONS

The analysis conducted is presented only as an example of the type of analysis possible with KHAFJI. The input parameters for the analysis were chosen arbitrarily and should not be taken as valid data. Using valid values for such parameters as mine P_k, movement degradation factor, and red kill rates may significantly change the outcome of the analysis.

V. CONCLUSIONS

A. SUMMARY

This thesis presented KHAFJI, a high resolution simulation which models minefield breaching tactics. As an example of the type of analysis that can be performed using KHAFJI, the use of artillery and/or multiple lanes in breaching a minefield was analyzed.

KHAFJI is a high fidelity combat simulation written in SIMSCRIPT II.5 with SIMGRAPHICS I. Employing user input parameters which define a minefield scenario, the model generates output which enables the user to compare various tactical options available to a maneuver commander in crossing a minefield. By using menu driven input screens, the user has a choice of multiple crossing lanes, size of crossing force, distribution of forces upon crossing lanes, multiple mine belts, and use of indirect fires against the minefield.

KHAFJI maintains a complete audit trail of significant events affecting the crossing unit. The model records the location of each mine, destruction of any mine by artillery, encounter of a mine by a member of the crossing unit, disablement of any member of the crossing unit by any means, final status of each member of the crossing unit, total rounds of artillery fired, total number of unit members successfully transiting the minefield, and unit crossing time.

Using SIMGRAPHICS I software, KHAFJI displays the minefield and the unit as it crosses the minefield. KHAFJI depicts each mine, each member of the crossing unit, and each impacting artillery round. The graphics provided by KHAFJI allows the user to see the crossing as it unfolds, thereby, reinforcing user confidence in the resultant data. When running multiple replications, graphics can be turned off to speed processing.

KHAFJI has been tested on desktop 286 and 386, and IBM 3033 (main frame) computers. The model is transportable (a feature of SIMSCRIPT) and is flexible in allowing a wide range of input parameters, which enable analysts to tailor scenarios to suit their needs.

B. POTENTIAL USES

KHAFJI can provide insights into fundamental questions involving minefield tactics such as: given competing tactical options in crossing a minefield, which options yield

maximum survivors; which options yield minimum crossing time; are multiple lanes and/or artillery useful in crossing a minefield; given a proposed anti-mine munition, is it effective in reducing the lethality of the minefield; given competing friendly defensive options, which options are most effective in terms of enemy casualties and maximum time for enemy to breach the minefield?

C. RECOMMENDATIONS

KHAFJI's modular engineering facilitates modifications. The addition of varying types of land mines is recommended. Currently only pressure mines are actually modeled within the mine encounter algorithms. Magnetic, contact, WAM and anti-personnel landmines should be added.

Additionally, a sophisticated direct fire algorithm is needed to further analyze the effects of direct fire on multiple lane options. A terrain model should be incorporated to better model the effects of artillery fires on terrain trafficability (affects mobility). Red artillery fires, algorithms currently contained within the model, should be "worked" into a battle scenario. A counter-battery algorithm should be developed to fight a "total battle."

Most importantly, KHAFJI should be merged with the model created by Anderson [Ref. 6]. This will allow the user to analyze minefield breaching tactics which include use of artillery, multiple lanes, and vehicular mounted mine plows.

APPENDIX A. DATA ANALYSIS

A. MOE - SURVIVORS

1 Lane:

Empirical Comparison of Marginal Distributions

X LABEL : LEVELS OF ARTILLERY: 0 20 40 60 80 RDS PER LANE

Y LABEL : NUMBER SURVIVORS

POPULATION NO. OF				-PERCENTILES-		
NUM	PNTS	YMEAN	YSDEV	0.25	0.5	0.75
1	5	2.6	0.54772	2	3	3
2	5	3	1	2	3	4
3	5	4	0.70711	4	4	4
4	5	3	2	2	4	4
5	5	3.4	0.54772	3	3	4

POOLED STANDARD DEVIATION ESTIMATE: 1.1045

Bartlett's Test for Homogeneity of Variance

F TEST STATISTIC : 2.4878

DEGREES OF FREEDOM: 4 600

SIGNIFICANCE (P-VALUE) LEVEL: 0.042395

A P-VALUE THAT EXCEEDS $\alpha = 0.05$ SIGNIFICANCE LEVEL
MAY INDICATE EITHER NON-NORMALITY OR UNEQUAL VARIANCES

Cases: 1 Lane with 0,20,40,60,80 Rounds of Artillery
(FIVE TREATMENTS)

0 RDS	2O RDS	40 RDS	60 RDS	80 RDS
RUN1	RUN1	RUN1	RUN1	RUN1
RUN2	RUN2	RUN2	RUN2	RUN2
RUN3	RUN3	RUN3	RUN3	RUN3
RUN4	RUN4	RUN4	RUN4	RUN4

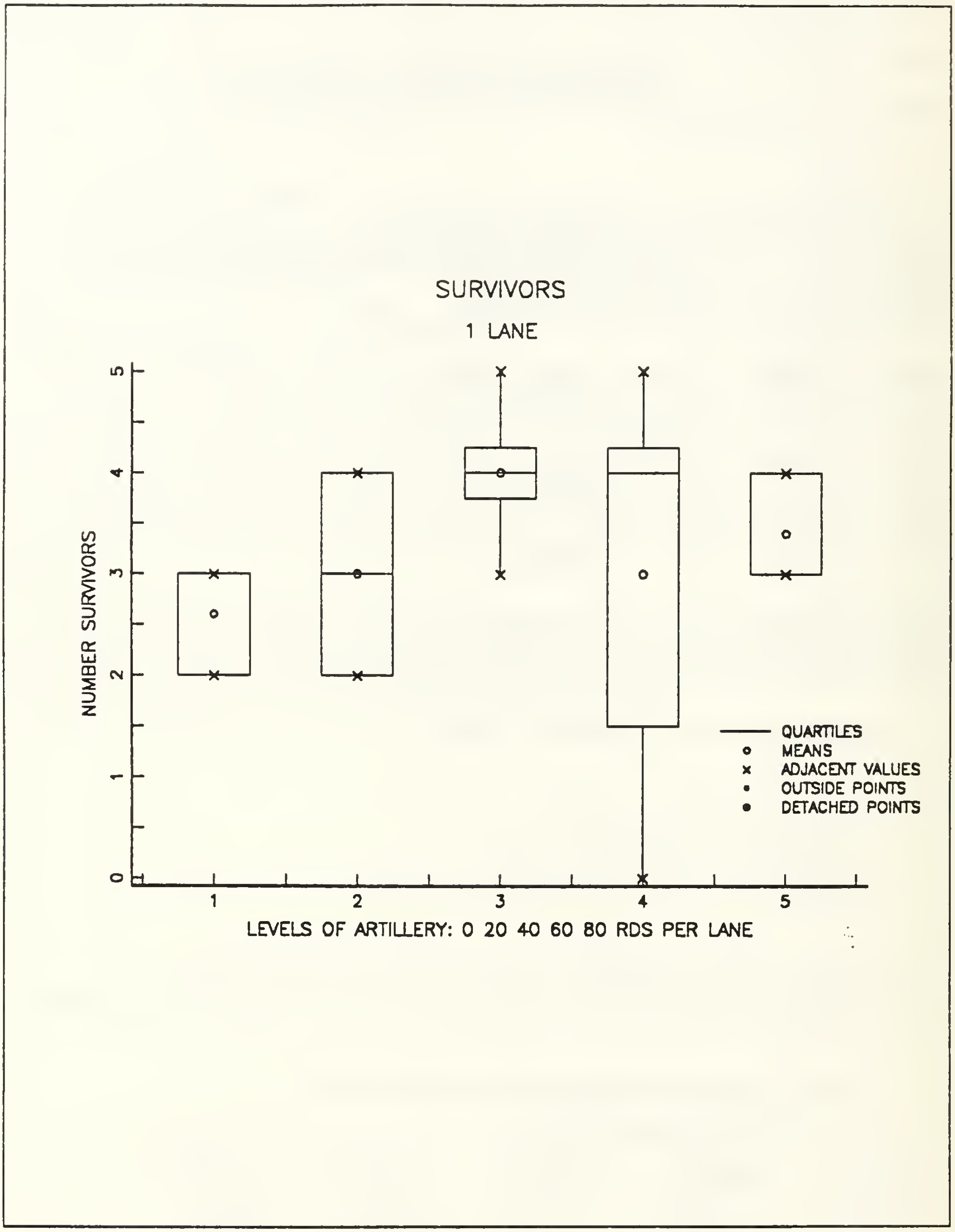

Figure 13. Box Plot - SURVIVORS - 1 Lane

RUN5 RUN5 RUN5 RUN5 RUN5

H_0: ALL RUN DISTRIBUTIONS FOR SURVIVORS ARE EQUAL
H_A: AT LEAST TWO OF THE RUN DISTRIBUTIONS DIFFER
$\alpha = 0.05$

(1) Analysis of Variance

SOURCE	SUM OF SQUARES	DF	MEAN SQUARE	F	SIG. LEVEL
BETWEEN	5.6	4	1.4	1.1475	0.36313
WITHIN	24.4	20	1.22		
TOTAL	30	24			

(2) Kruskal-Wallis Nonparametric Test

K-W STATISTIC (H): 5.9742
DEGREES OF FREEDOM: 4
ASYMPTOTIC SIGNIFICANCE (P-VALUE) LEVEL: 0.20108

SIGNIFICANCE LEVEL IS BASED ON CHI-SQUARE APPROXIMATION

<u>2 Lanes</u>

Empirical Comparison of Marginal Distributions

X LABEL : LEVELS OF ARTILLERY: 0 20 40 60 80 RDS PER LANE
Y LABEL : NUMBER SURVIVORS

POPULATION NUM	NO. OF PNTS	YMEAN	YSDEV	-PERCENTILES- 0.25	0.5	0.75
1	5	1.8	0.83666	1	2	2
2	5	3	1.2247	2	3	3
3	5	5.6	0.54772	5	6	6
4	5	5.4	0.54772	5	5	6
5	5	6.4	0.89443	6	7	7

POOLED STANDARD DEVIATION ESTIMATE: 0.84853

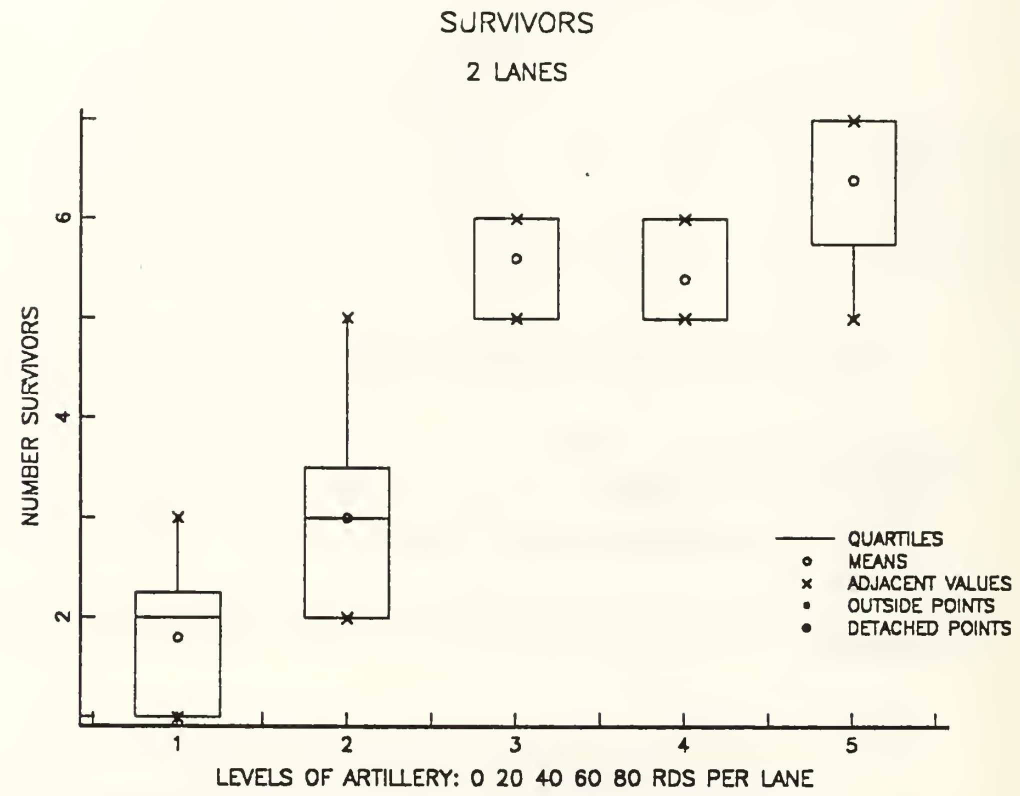

Figure 14. Box Plot - SURVIVORS - 2 Lanes

Bartlett's Test for Homogeneity of Variance

F TEST STATISTIC : 0.85377
DEGREES OF FREEDOM: 4 600
SIGNIFICANCE (P-VALUE) LEVEL: 0.49153

A P-VALUE THAT EXCEEDS $\alpha = 0.05$ SIGNIFICANCE LEVEL
MAY INDICATE EITHER NON-NORMALITY OR UNEQUAL VARIANCES

Cases: 2 Lanes with 0,20,40,60,80 Rounds of Artillery
(FIVE TREATMENTS)

0 RDS	20 RDS	40 RDS	60 RDS	80 RDS
RUN1	RUN1	RUN1	RUN1	RUN1
RUN2	RUN2	RUN2	RUN2	RUN2
RUN3	RUN3	RUN3	RUN3	RUN3
RUN4	RUN4	RUN4	RUN4	RUN4
RUN5	RUN5	RUN5	RUN5	RUN5

H_0: ALL RUN DISTRIBUTIONS FOR SURVIVORS ARE EQUAL
H_A: AT LEAST TWO OF THE RUN DISTRIBUTIONS DIFFER
$\alpha = 0.05$

(3) Kruskal-Wallis Nonparametric Test

K-W STATISTIC (H): 18.751
DEGREES OF FREEDOM: 4
ASYMPTOTIC SIGNIFICANCE (P-VALUE) LEVEL: 0.00087953
SIGNIFICANCE LEVEL IS BASED ON CHI-SQUARE APPROXIMATION

3 Lanes

Empirical Comparison of Marginal Distributions

X LABEL : LEVELS OF ARTILLERY: 0 20 40 60 80 RDS PER LANE
Y LABEL : NUMBER SURVIVORS

POPULATION NUM	NO. OF PNTS	YMEAN	YSDEV	-PERCENTILES- 0.25	0.5	0.75

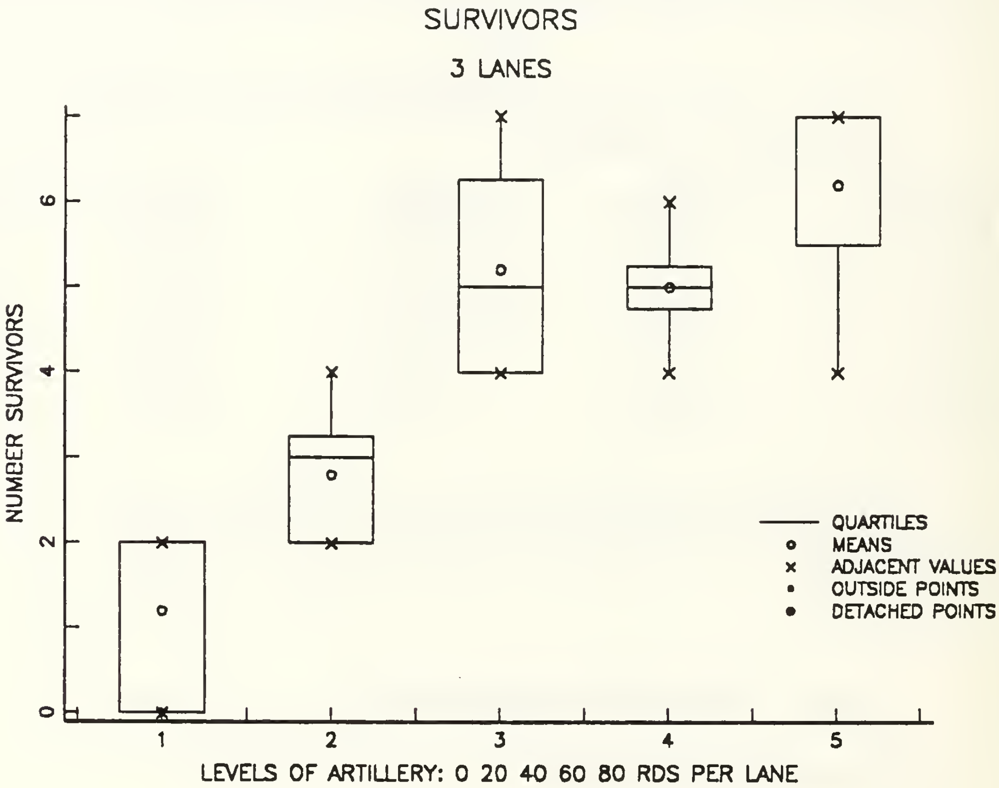

Figure 15. Box Plot - SURVIVORS - 3 Lanes

1	5	1.2	1.0954	0	2	2
2	5	2.8	0.83666	2	3	3
3	5	5.2	1.3038	4	5	6
4	5	5	0.70711	5	5	5
5	5	6.2	1.3038	6	7	7

POOLED STANDARD DEVIATION ESTIMATE: 1.077

Bartlett's Test for Homogeneity of Variance

F TEST STATISTIC : 0.49698
DEGREES OF FREEDOM: 4 600
SIGNIFICANCE (P-VALUE) LEVEL: 0.73798

A P-VALUE THAT EXCEEDS $\alpha = 0.05$ SIGNIFICANCE LEVEL
MAY INDICATE EITHER NON-NORMALITY OR UNEQUAL VARIANCES

Cases: 3 Lanes with 0,20,40,60,80 Rounds of Artillery
(FIVE TREATMENTS)

0 RDS	2O RDS	40 RDS	60 RDS	80 RDS
RUN1	RUN1	RUN1	RUN1	RUN1
RUN2	RUN2	RUN2	RUN2	RUN2
RUN3	RUN3	RUN3	RUN3	RUN3
RUN4	RUN4	RUN4	RUN4	RUN4
RUN5	RUN5	RUN5	RUN5	RUN5

H_0: ALL RUN DISTRIBUTIONS FOR SURVIVORS ARE EQUAL
H_A: AT LEAST TWO OF THE RUN DISTRIBUTIONS DIFFER
$\alpha = 0.05$

(4) Kruskal-Wallis Nonparametric Test

K-W STATISTIC (H): 18.656
DEGREES OF FREEDOM: 4
ASYMPTOTIC SIGNIFICANCE (P-VALUE) LEVEL: 0.00091791

SIGNIFICANCE LEVEL IS BASED ON CHI-SQUARE APPROXIMATION

B. MOE - BATTLE TIME

1 Lane

Empirical Comparison of Marginal Distributions

X LABEL : LEVELS OF ARTILLERY: 0 20 40 60 80 RDS PER LANE

Y LABEL : BATTLE TIME

POPULATION NUM	NO. OF PNTS	YMEAN	YSDEV	PERCENTILES 0.25	0.5	0.75
1	5	35.566	3.2077	33.91	34	35.17
2	5	43.618	4.7692	41.14	41.59	44.47
3	5	41.282	3.5353	41.5	41.61	41.8
4	5	45	4.0087	44.7	45.4	45.9
5	5	55.76	0.65038	55.4	55.6	55.6

POOLED STANDARD DEVIATION ESTIMATE: 3.5221

Bartlett's Test for Homogeneity of Variance

F TEST STATISTIC : 2.4727

DEGREES OF FREEDOM: 4 600

SIGNIFICANCE (P-VALUE) LEVEL: 0.043453

A P-VALUE THAT EXCEEDS $\alpha = 0.05$ SIGNIFICANCE LEVEL
MAY INDICATE EITHER NON-NORMALITY OR UNEQUAL VARIANCES

Cases: 1 Lane with 0,20,40,60,80 Rounds of Artillery
(FIVE TREATMENTS)

0 RDS	2O RDS	40 RDS	60 RDS	80 RDS
RUN1	RUN1	RUN1	RUN1	RUN1
RUN2	RUN2	RUN2	RUN2	RUN2
RUN3	RUN3	RUN3	RUN3	RUN3
RUN4	RUN4	RUN4	RUN4	RUN4
RUN5	RUN5	RUN5	RUN5	RUN5

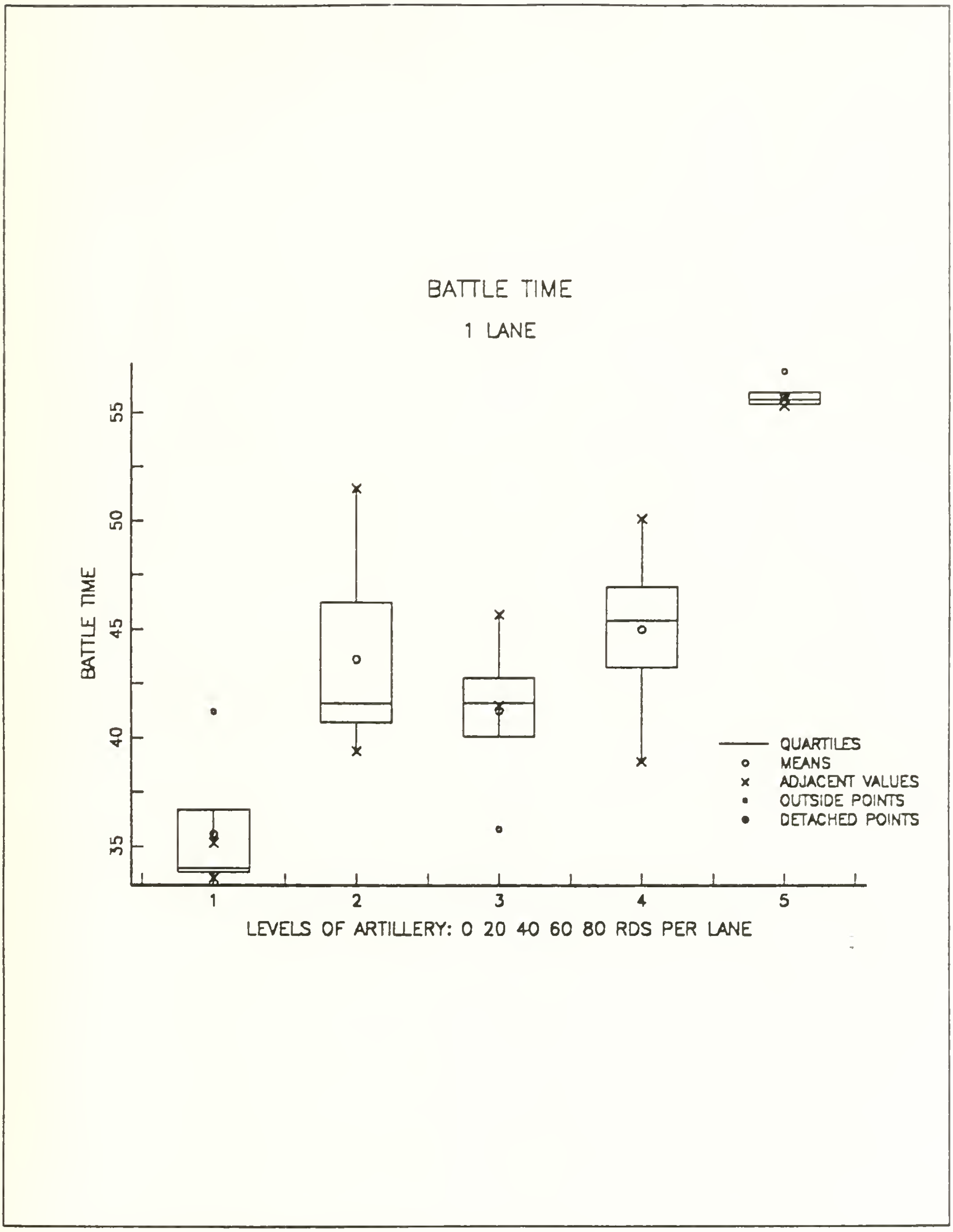

Figure 16. Box Plot - BATTLE TIME - 1 Lane

H_0: ALL RUN DISTRIBUTIONS FOR BATTLE TIME ARE EQUAL
H_A: AT LEAST TWO OF THE RUN DISTRIBUTIONS DIFFER
$\alpha = 0.05$

(1) Analysis of Variance

SOURCE	SUM OF SQUARES	DF	MEAN SQUARE	F	SIG. LEVEL
BETWEEN	1088.3	4	272.08	21.933	4.4468E-7
WITHIN	248.11	20	12.405		
TOTAL	1336.4	24			

(2) Kruskal-Wallis Nonparametric Test

K-W STATISTIC (H): 17.678
DEGREES OF FREEDOM: 4
ASYMPTOTIC SIGNIFICANCE (P-VALUE) LEVEL: 0.001426
SIGNIFICANCE LEVEL IS BASED ON CHI-SQUARE APPROXIMATION

2 Lanes

Empirical Comparison of Marginal Distributions

X LABEL : LEVELS OF ARTILLERY: 0 20 40 60 80 RDS PER LANE
Y LABEL : BATTLE TIME

POPULATION NUM	NO. OF PNTS	YMEAN	YSDEV	----PERCENTILES---- 0.25	0.5	0.75
1	5	11.466	2.4338	11.9	12.02	12.33
2	5	19.51	3.7978	16.4	21.9	22.4
3	5	21.4	1.1247	20.5	21.4	21.9
4	5	24.7	0.83367	24.4	24.4	24.7
5	5	27.6	0.88034	26.9	27.2	28.5

POOLED STANDARD DEVIATION ESTIMATE: 2.1486

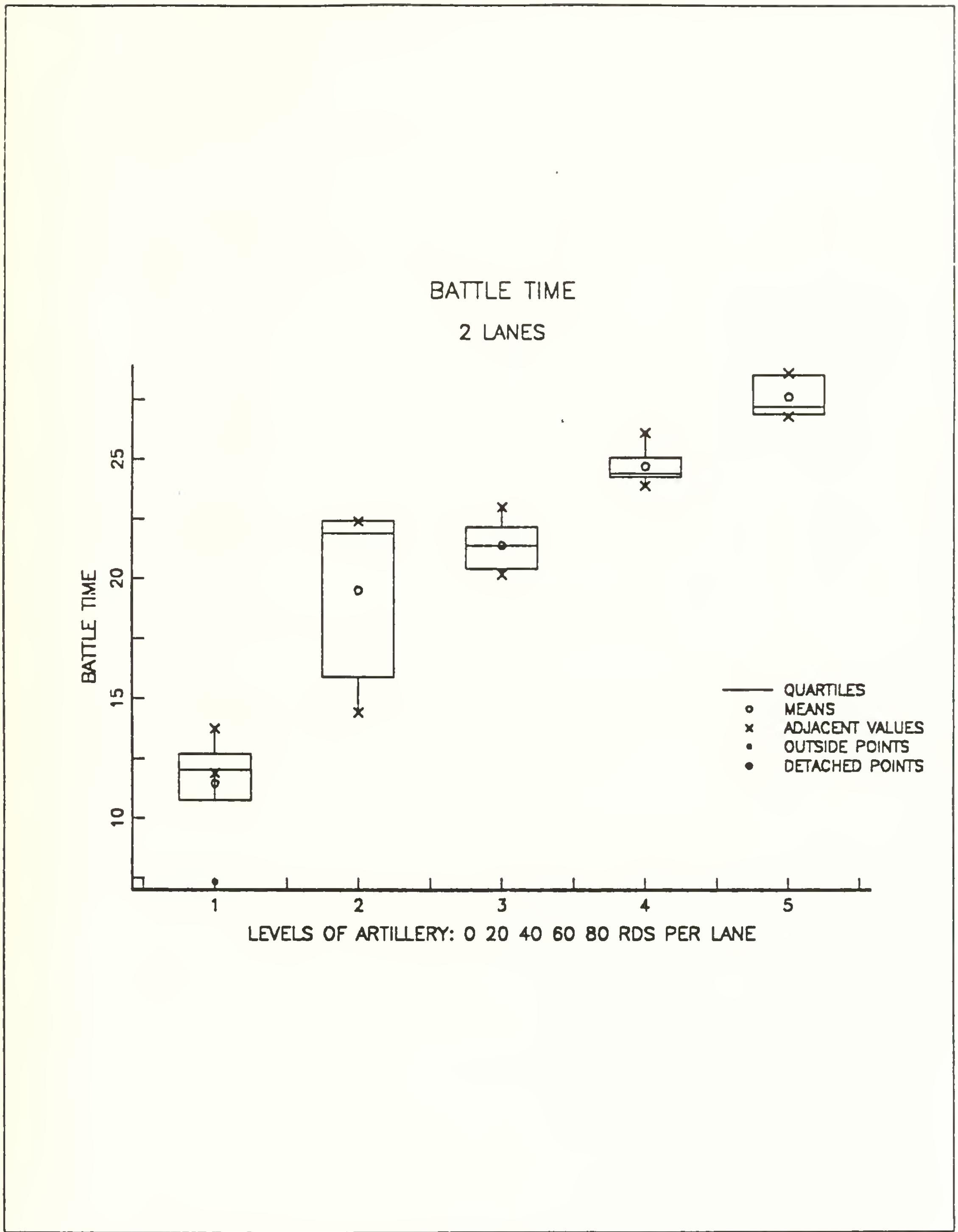

Figure 17. Box Plot - BATTLE TIME - 2 Lanes

Bartlett's Test for Homogeneity of Variance

F TEST STATISTIC : 3.3088
DEGREES OF FREEDOM: 4
SIGNIFICANCE (P-VALUE) LEVEL: 0.010735

A P-VALUE THAT EXCEEDS $\alpha = 0.05$ SIGNIFICANCE LEVEL
MAY INDICATE EITHER NON-NORMALITY OR UNEQUAL VARIANCES

Cases: 2 Lanes with 0,20,40,60,80 Rounds of Artillery
(FIVE TREATMENTS)

0 RDS	20 RDS	40 RDS	60 RDS	80 RDS
RUN1	RUN1	RUN1	RUN1	RUN1
RUN2	RUN2	RUN2	RUN2	RUN2
RUN3	RUN3	RUN3	RUN3	RUN3
RUN4	RUN4	RUN4	RUN4	RUN4
RUN5	RUN5	RUN5	RUN5	RUN5

H_0: ALL RUN DISTRIBUTIONS FOR BATTLE TIME ARE EQUAL
H_A: AT LEAST TWO OF THE RUN DISTRIBUTIONS DIFFER
$\alpha = 0.05$

(3) Analysis of Variance

SOURCE	SUM OF SQUARES	DF	MEAN SQUARE	F	SIG. LEVEL
BETWEEN	752.53	4	188.13	40.754	2.4062E-9
WITHIN	92.326	20	4.6163		
TOTAL	844.86	24			

(4) Kruskal-Wallis Nonparametric Test

K-W STATISTIC (H): 21.956
DEGREES OF FREEDOM: 4
ASYMPTOTIC SIGNIFICANCE (P-VALUE) LEVEL: 0.0002045

SIGNIFICANCE LEVEL IS BASED ON CHI-SQUARE APPROXIMATION

3 Lanes

Empirical Comparison of Marginal Distributions

X LABEL : LEVELS OF ARTILLERY: 0 20 40 60 80 RDS PER LANE

Y LABEL : BATTLE TIME

POPULATION NUM	NO. OF PNTS	YMEAN	YSDEV	----PERCENTILES---- 0.25	0.5	0.75
1	5	5.168	1.999	3.69	3.76	7.22
2	5	13.8	2.5189	14.7	15	15
3	5	13.4	3.9414	9.7	14.5	15
4	5	16.5	0.41231	16.2	16.4	16.5
5	5	19.2	0.21213	19.2	19.2	19.2

POOLED STANDARD DEVIATION ESTIMATE: 2.2843

Bartlett's Test for Homogeneity of Variance

F TEST STATISTIC : 6.7571

DEGREES OF FREEDOM: 4 600

SIGNIFICANCE (P-VALUE) LEVEL: 0.000025305

A P-VALUE THAT EXCEEDS $\alpha = 0.05$ SIGNIFICANCE LEVEL
MAY INDICATE EITHER NON-NORMALITY OR UNEQUAL VARIANCES

Cases: 3 Lanes with 0,20,40,60,80 Rounds of Artillery

(FIVE TREATMENTS)

0 RDS	2O RDS	40 RDS	60 RDS	80 RDS
RUN1	RUN1	RUN1	RUN1	RUN1
RUN2	RUN2	RUN2	RUN2	RUN2
RUN3	RUN3	RUN3	RUN3	RUN3
RUN4	RUN4	RUN4	RUN4	RUN4
RUN5	RUN5	RUN5	RUN5	RUN5

H_0: ALL RUN DISTRIBUTIONS FOR BATTLE TIME ARE EQUAL

H_A: AT LEAST TWO OF THE RUN DISTRIBUTIONS DIFFER

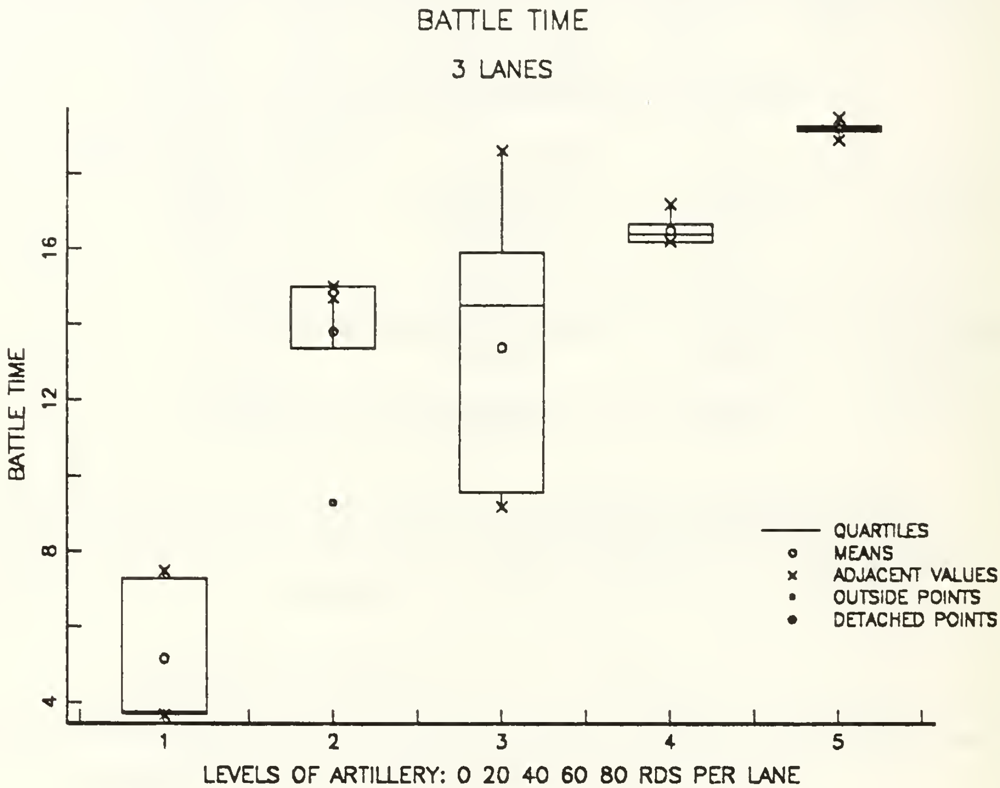

Figure 18. Box Plot - BATTLE TIME - 3 Lanes

$\alpha = 0.05$

(5) Analysis of Variance

SOURCE	SUM OF SQUARES	DF	MEAN SQUARE	F	SIG. LEVEL
BETWEEN	554.74	4	138.68	26.577	9.3294E-8
WITHIN	104.36	20	5.2182		
TOTAL	659.1	24			

(6) Kruskal-Wallis Nonparametric Test

K-W STATISTIC (H): 20.799
DEGREES OF FREEDOM: 4
ASYMPTOTIC SIGNIFICANCE LEVEL: 0.0003471

SIGNIFICANCE LEVEL IS BASED ON CHI-SQUARE APPROXIMATION

APPENDIX B. RUN DATA

A. WITHOUT ARTILLERY

1 Lane

#	SURVIVORS	BATTLE TIME(MINUTES)	KILLS MINE	KILLS DIRECT FIRE
1	2	41.2	3	2
2	3	33.91	2	2
3	3	33.55	3	1
4	3	34	4	0
5	2	35.17	4	3

2 Lanes

#	SURVIVORS	BATTLE TIME(MINUTES)	KILLS MINE	KILLS DIRECT FIRE
1	1	13.76	6	0
2	2	12.02	5	0
3	3	12.33	4	0
4	2	7.32	5	0
5	1	11.9	6	0

3 Lanes

#	SURVIVORS	BATTLE TIME(MINUTES)	KILLS MINE	KILLS DIRECT FIRE
1	1	3.76	5	0
2	2	7.22	7	0
3	3	3.68	5	0
4	2	3.69	5	0
5	1	7.49	7	0

B. 20 ROUNDS OF ARTILLERY

1 Lane

#	SURVIVORS	BATTLE TIME(MINUTES)	KILLS MINE	KILLS DIRECT FIRE
1	2	51.5	3	2
2	2	39.39	2	3

3	4	41.14	1	2
4	4	44.47	1	2
5	3	41.57	2	2

2 Lanes

			KILLS	
#	SURVIVORS	BATTLE TIME(MINUTES)	MINE	DIRECT FIRE
1	3	22.4	4	0
2	3	16.4	4	0
3	5	22.4	2	0
4	2	14.45	5	0
5	2	21.9	5	0

3 Lanes

			KILLS	
#	SURVIVORS	BATTLE TIME(MINUTES)	MINE	DIRECT FIRE
1	4	15	3	0
2	3	15	4	0
3	2	9.3	5	0
4	3	14.7	4	0
5	2	15	5	0

C. 40 ROUNDS OF ARTILLERY

1 Lane

			KILLS	
#	SURVIVORS	BATTLE TIME(MINUTES)	MINE	DIRECT FIRE
1	5	41.8	0	2
2	4	41.61	1	2
3	4	35.8	1	2
4	4	45.7	1	2
5	3	41.5	2	2

2 Lanes

			KILLS	
#	SURVIVORS	BATTLE TIME(MINUTES)	MINE	DIRECT FIRE
1	5	23	4	0
2	6	20.2	4	0
3	5	21.9	2	0

4	6	20.5	5	0
5	6	21.4	5	0

3 Lanes

			KILLS	
#	SURVIVORS	BATTLE TIME(MINUTES)	MINE	DIRECT FIRE
1	4	15	3	0
2	4	9.2	3	0
3	7	14.5	2	0
4	7	9.7	0	0
5	6	18.6	1	0

D. 60 ROUNDS OF ARTILLERY

1 Lane

			KILLS	
#	SURVIVORS	BATTLE TIME(MINUTES)	MINE	DIRECT FIRE
1	4	38.9	1	3
2	4	50.1	0	3
3	0	44.7	2	5
4	5	45.4	0	2
5	2	45.9	1	2

2 Lanes

			KILLS	
#	SURVIVORS	BATTLE TIME(MINUTES)	MINE	DIRECT FIRE
1	5	24.4	2	0
2	5	24.4	2	0
3	6	24.7	1	0
4	6	26.1	1	0
5	5	23.9	2	0

3 Lanes

			KILLS	
#	SURVIVORS	BATTLE TIME(MINUTES)	MINE	DIRECT FIRE
1	5	17.2	2	0
2	5	16.2	2	0
3	4	16.4	3	0
4	5	16.2	2	0

5	6	16.5	1	0

E. 80 ROUNDS OF ARTILLERY

1 Lane

			KILLS	
#	SURVIVORS	BATTLE TIME(MINUTES)	MINE	DIRECT FIRE
1	3	55.6	1	3
2	3	56.9	0	4
3	4	55.4	0	3
4	3	55.3	1	3
5	4	55.6	0	3

2 Lanes

			KILLS	
#	SURVIVORS	BATTLE TIME(MINUTES)	MINE	DIRECT FIRE
1	7	26.9	0	0
2	7	28.5	0	0
3	7	26.8	0	0
4	6	28.6	1	0
5	5	27.2	2	0

3 Lanes

			KILLS	
#	SURVIVORS	BATTLE TIME(MINUTES)	MINE	DIRECT FIRE
1	7	19.2	0	0
2	4	19.2	3	0
3	6	19.5	1	0
4	7	18.9	0	0
5	7	19.2	0	0

APPENDIX C. SAMPLE OUTPUT

MINE DENSITY OF BELT 1 : .250 MINES PER METER

MINE #	EASTING	NORTHING
------	-------	--------
10001	NX 56001.3	78063.0
10002	NX 56005.8	78043.9
10003	NX 56009.7	78047.1
10004	NX 56015.2	78050.7
10005	NX 56018.0	78061.7
10006	NX 56021.3	78040.2
10007	NX 56025.0	78039.8
10008	NX 56029.2	78046.9
10009	NX 56035.1	78047.7
10010	NX 56036.9	78050.3
10011	NX 56041.6	78038.7
10012	NX 56046.2	78046.0
10013	NX 56050.2	78050.5
10014	NX 56053.5	78048.3

10015	NX 56058.2	78040.3
10016	NX 56062.4	78047.2
10017	NX 56066.2	78049.1
10018	NX 56069.6	78046.7
10019	NX 56073.6	78048.6
10020	NX 56078.2	78050.1
10021	NX 56081.9	78059.4
10022	NX 56084.6	78050.7
10023	NX 56089.6	78048.4
10024	NX 56095.4	78049.1
10025	NX 56098.5	78050.9
10026	NX 56101.3	78051.6
10027	NX 56105.1	78044.1
10028	NX 56109.8	78045.4
10029	NX 56113.6	78057.5
10030	NX 56118.8	78053.7
10031	NX 56123.0	78037.1

10032	NX	56125.2	78047.4
10033	NX	56130.6	78028.4
10034	NX	56133.3	78050.5
10035	NX	56138.2	78063.7
10036	NX	56141.6	78039.9
10037	NX	56145.7	78049.5
10038	NX	56150.6	78039.2
10039	NX	56153.6	78046.6
10040	NX	56157.6	78059.3
10041	NX	56162.2	78052.3
10042	NX	56167.0	78053.0
10043	NX	56169.2	78027.7
10044	NX	56174.3	78048.8
10045	NX	56178.6	78043.5
10046	NX	56182.0	78058.4
10047	NX	56186.2	78043.5
10048	NX	56189.9	78071.2

10049 NX 56193.7 78046.1

10050 NX 56197.5 78049.5

MINEFIELD CONTAINS 50 MINES

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10014 NX 56053.5 78048.3

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10018 NX 56069.6 78046.7

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10019 NX 56073.6 78048.6

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10018 NX 56069.6 78046.7

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10015 NX 56058.2 78040.3

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10014 NX 56053.5 78048.3

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10013 NX 56050.2 78050.5

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10016 NX 56062.4 78047.2

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10015 NX 56058.2 78040.3

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10011 NX 56041.6 78038.7

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10008 NX 56029.2 78046.9

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10009 NX 56035.1 78047.7

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10001 NX 56001.3 78063.0

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10022 NX 56084.6 78050.7

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10030 NX 56118.8 78053.7

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10021 NX 56081.9 78059.4

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10033 NX 56130.6 78028.4

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10024 NX 56095.4 78049.1

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10025 NX 56098.5 78050.9

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10024 NX 56095.4 78049.1

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10025 NX 56098.5 78050.9

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10026 NX 56101.3 78051.6

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10029 NX 56113.6 78057.5

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10030 NX 56118.8 78053.7

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10030 NX 56118.8 78053.7

MINE DESTROYED BY BLUE HOWITZER AT GRID:

10026 NX 56101.3 78051.6

MINES DESTROYED 26 MINES REMAINING 24

ELEMENT 2 DESTROYED BY MINE 10012 AT 3.51

ELEMENT 3 KILLED BY DIRECT FIRE AT 10.45

ELEMENT	STATUS	TIME TO DEATH
7	THROUGH	32.05
6	THROUGH	28.93
5	THROUGH	54.07
4	THROUGH	27.69
1	THROUGH	23.02
2	OBSTACL	34.61
3	OBSTACL	10.44

FOR RUN 1 BATTLE TIME: 16.11

LANE 2	MEAN	VARIANCE
ARTY YES	----	--------
NUMBER SURVIVORS	5.0	0.
BATTLE TIME	16.1	0.

ROUNDS FIRED BY BLUE 160

ROUNDS FIRED BY RED 0

APPENDIX D. PROGRAM LISTING

A. PREAMBLE

```
PREAMBLE

    NORMALLY MODE IS INTEGER

    PERMANENT ENTITIES
       EVERY LANE HAS  ''CROSSING LANE 9 MAX
       A START.X,      ''LANE ENTRY POINT
       A N.ELEMENT     ''# BREACHERS EACH LANE
       DEFINE N.ELEMENT AS AN INTEGER VARIABLE
       DEFINE START.X AS A REAL VARIABLE

       EVERY MINE.BELT HAS ''MINES LOCATED WITHIN
       A N.MINES,          ''# MINE EACH BELT
       A DEPTH,            ''DISTANCE INTO MINEFIELD
       A DENSITY,          ''METERS PER MINE
       A MINE.B.TYPE       ''MINE TYPES HOMOGENEOUS
                   ''WITHIN BELTS
       DEFINE N.MINES,MINE.B.TYPE AS INTEGER VARIABLES
       DEFINE  DEPTH, DENSITY AS REAL VARIABLES

       EVERY WEAPON HAS    ''INDIRECT FIRE WEAPONS
       A WEAPON.TYPE,      ''HOWITZER OR CHOICE
       A LETHAL.RADIUS, ''LETHAL BURST RADIUS OF ROUND
       A ERROR.X,          ''IMPACT ERROR EASTING
       A ERROR.Y,          ''IMPACT ERROR NORTHING
       A WEAPON.UNIT,      ''BLUE OR RED
       MAY BELONG TO THE BLUE.TGT.LIST ''RED TARGETING
       DEFINE LETHAL.RADIUS,ERROR.X,ERROR.Y
       AS REAL VARIABLES
       DEFINE WEAPON.TYPE,WEAPON.UNIT AS TEXT VARIABLES
```

```
TEMPORARY ENTITIES
    EVERY MINE HAS
    A RADIUS,
    A MINE.STATUS,      ''ACTIVE OR DESTROYED
    A MINE.TYPE,        ''PRESSURE, CONTACT, ETC.
    A MINE.X,           ''MINE EASTING
    A MINE.Y,           ''MINE NORTHING
    A MINE.NUMBER,      ''MINE TARGET #
    A DEST.RADIUS       ''MINE LETHAT RADIUS
    AND BELONGS TO THE MINE.FIELD ''GROUPING
    DEFINE MINE.TYPE AS AN INTEGER VARIABLE
    DEFINE DEST.RADIUS,MINE.X,MINE.Y,RADIUS
    AS REAL VARIABLES
    DEFINE MINE.STATUS AS AN TEXT VARIABLES
    DEFINE MINE.FIELD AS A SET RANKED BY LOW MINE.Y

    EVERY ELEMENT HAS   ''BREACHER
    A ELEMENT.X,        ''BREACHER EASTING
    A ELEMENT.Y,        ''BREACHER NORTHING
    A ELEMENT.STATUS,   ''ACTIVE,BYPASS,THROUGH,
                ''OR OBSTACLE(DISABLED)
    A ELEMENT.TYPE,
    A SPEED,
    A WIDTH,
    A ELEMENT.NUM,
    A DEATH.TIME, ''TIME TO DEATH BY DIRECT FIRE
    A ELEM.RADIUS,
    A TRACK.WIDTH,
    MAY BELONG TO THE RED.TGT.LIST,AND MAY BELONG
    TO THE OBSTACLE.LIST ''IF DISABLED
    DEFINE ELEMENT.NUM,ELEMENT.TYPE AS INTEGER VARIABLES
    DEFINE ELEMENT.X,ELEMENT.Y,SPEED,WIDTH,DEATH.TIME
    AS REAL VARIABLES
```

```
    DEFINE ELEMENT.STATUS AS A TEXT VARIABLE
    DEFINE RED.TGT.LIST AS A SET RANKED BY LOW ELEMENT.X
    DEFINE OBSTACLE.LIST AS A SET RANKED BY LOW ELEMENT.X
   ''V-POINTER FOR ELEMENT (TEMP ENTITIES ARE UNORDERED)
    DEFINE V AS A 1-DIMENSIONAL POINTER ARRAY


GRAPHIC ENTITIES INCLUDE MAP, ''DISPLAY BACKGROUND
XPLODE, ''ARTY IMPACTS
DEAD.TANK  ''OBSTACLE
DEFINE DT AS A 1-DIMENSIONAL POINTER ARRAY ''ICN POINTER
DEFINE XB AS A 1-DIMENSIONAL POINTER ARRAY " "     "
DEFINE XR AS A 1-DIMENSIONAL POINTER ARRAY " "     "
DYNAMIC GRAPHIC ENTITIES INCLUDE MINE, ''DISPLAYS MINE
ELEMENT                          " DISPLAYS BREACHERS

DEFINE GR.ZONE,          ''GRID ZONE DESIGNATION
FIELD.ID,              ''INPUT FORM POINTER
GO,                  ''YES TO USE ARTILLERY
GRAPHICS,              ''YES TO DISPLAY GRAPHICS
CONTINUE               ''YES FOR ANOTHER ITERATION
AS TEXT VARIABLES

DEFINE MF.WIDTH,   ''MINEFIELD WIDTH
GR.BASE.E, ''10,000 METER DESIGNATION NORTHING
GR.BASE.N,        "         "      EASTING
COMM.TIME,   ''USED TO COMPUTE BATTLE TIME
E.WIDTH,     ''BREACHER WIDTH
TEMP.RADIUS, ''MINE RADIUS
ONE.ONE,ONE.TWO,ONE.THREE,TWO.ONE, ''PK TABLE VALUES
TWO.TWO,TWO.THREE,THREE.ONE,THREE.TWO,THREE.THREE" "
BATTLE.TIME,  ''ELAPSED TIME FOR CROSSING
N.SURVIVOR,   ''# SURVIVING CROSSING
E.LENGTH,     ''BREACHER LENGTH
```

```
KILL.RATE      ''LANDCHESTERIAN KILL COEFFICIENT
AS REAL VARIABLES

DEFINE INPLACED,      ''COUNTER FOR 1 MINE INPLACED
INITIAL,              ''INTIAL # MINES INPLACED
N.VOLLEYS,            ''# VOLLEYS TO FIRE INTO MINEFIELD
NUM.CP,               ''# CHECKPOINTS FOR NAVIGATION
TOT.ELEMENT,          ''TOTAL BREACHERS
MAX.DISTANCE,         ''LENGTH OF MINEFIELD
N.RUN,                ''# ITERATIONS TO PERFORM (RUNS)
N.BLUE.ROUND,         ''# ARTILLERY ROUNDS FIRED BY BLUE
N.RED.ROUND           ''         ''         RED
AS INTEGER VARIABLES
                                ''GRAPHICAL POINTERS
DEFINE INPUT.FORM,PK.FORM,DATA.FORM,DEVPTR, ''  ''
INPUT.FORM, INPUT3.FORM,INPUT4.FORM, INPUT5.FORM,''''
TANK.FORM AS POINTER VARIABLES

''TO HOLD GRAPHICS ON SCREEN
SUBSTITUTE THESE 5 LINES FOR ..MOUSE.PAUSE
CALL READLOC.R GIVEN 0,0,0
   YIELDING DUMMY.X,DUMMY.Y,DUMMY.V
   DUMMY.X = DUMMY.X
   DUMMY.Y = DUMMY.Y
   DUMMY.V = DUMMY.V

''TO CONVERT SYSTEM TIME TO MINUTES
DEFINE MINUTES TO MEAN DAYS
DEFINE SECONDS TO MEAN HOURS
DEFINE MILLISECONDS TO MEAN MINUTES

PROCESSES INCLUDE BLUE.ARTY.ATK.MINE.FIELD
  EVERY RED.ARTY.ATK HAS A TARGET.X, A TARGET.Y
  DEFINE TARGET.X, TARGET.Y AS REAL VARIABLES
```

```
    EVENT NOTICES
      EVERY MINE.ENCOUNTER HAS A LINE, AN ID, A MINE.ID,
      A M.NUMBER
      EVERY NEW.CP HAS A LINE, AN ID
      EVERY OBSTACLE.ENCOUNTER HAS A LINE, AN ID

    THE SYSTEM OWNS A MINE.FIELD, A BLUE.TGT.LIST,
    A RED.TGT.LIST,AN OBSTACLE.LIST

    TALLY TOT.MINES AS THE SUM OF N.MINES
    TALLY TOT.ACTIVE AS THE SUM OF INPLACED
    TALLY MEAN.SURVIVOR AS THE MEAN AND VAR.SURVIVOR AS THE
    VARIANCE OF N.SURVIVOR
    TALLY MEAN.BATTLE.TIME AS THE MEAN AND VAR.BATTLE.TIME
    AS THE VARIANCE OF BATTLE.TIME

    ''TO STORE RANDOM NUMBER SEEDS
    DEFINE S AS A REAL 1-DIMENSIONAL VARIABLE

    ''TO STORE CHECKPOINTS FOR EACH BREACHER
    DEFINE CP AS A REAL 2-DIMENSIONAL VARIABLE

    ''TO STORE PATHS THROUGH MINEFIEDL FOR EACH BREACHER
    DEFINE MOVEMENT.PLAN AS A REAL 3-DIMENSIONAL ARRAY

    ''TO STORE BYPASS ROUTES
    DEFINE BYPASS.MAP AS A REAL 3-DIMENSIONAL ARRAY

    ''TO STORE OBSTACLE IDENTIFICATIONS (WHICH BREACHER)
    DEFINE OBSTACLE.MAP AS A INTEGER 2-DIMENSIONAL ARRAY

    DEFINE LAY.MINE.FIELD AS REAL FUNCTIONS

END
```

B. MAIN

```
MAIN
      LET HOURS.V = 60
      LET MINUTES.V = 1000

      RESERVE S(*) AS 10
      RESERVE XB(*) AS 500
      RESERVE XR(*) AS 500

      CALL INITIALIZE

    FOR Z = 1 TO N.RUN DO

      COMM.TIME = TIME.V
      IF Z GT 1   ''TO CONDUCT MULTIPLE RUNS
       TOT.ELEMENT = N
       CALL RESTART
      ELSE
       N = TOT.ELEMENT
      ALWAYS

      CALL LAY.MINE.FIELD

      ''TO ALLOW FOR OPTIONAL GRAPHICS
      IF GRAPHICS EQ "YES" OR GRAPHICS EQ "yes"
       CREATE A MAP
       DISPLAY MAP WITH "MAP.ICN"
       CALL SET.DISPLAY
       FOR EACH MINE IN MINE.FIELD DO
         DISPLAY MINE
       LOOP
       FOR EACH ELEMENT IN RED.TGT.LIST DO
         DISPLAY ELEMENT
```

```
  LOOP
ALWAYS

''TO FIRE ARTILLERY AGAINST MINEFIELD
ACTIVATE A BLUE.ARTY.ATK.MINE.FIELD NOW

''TO START SIMULATION
CALL START.SIMO

''TO PRINT INTERMEDIATE RESULTS
SKIP 2 LINES
PRINT 2 LINES THUS
ELEMENT      STATUS      TIME TO DEATH
--------     ------      -------------

XX = 0
FOR EACH ELEMENT OF RED.TGT.LIST DO
PRINT 1 LINE WITH ELEMENT.NUM,ELEMENT.STATUS,
DEATH.TIME - COMM.TIME THUS
 ***        *******       ***.**
  IF ELEMENT.STATUS EQ "THROUGH"
   ADD 1 TO XX
  ALWAYS
LOOP


FOR EACH ELEMENT OF OBSTACLE.LIST DO
 PRINT 1 LINE WITH ELEMENT.NUM,ELEMENT.STATUS,DEATH.TIME -
COMM.TIME THUS
 ***        *******       ***.**
  REMOVE ELEMENT FROM OBSTACLE.LIST
  FILE ELEMENT IN RED.TGT.LIST
LOOP

''TO RESET/RESTART SIMULATION FOR NEXT RUN
```

```
RESET THE TOTALS OF INPLACED
BATTLE.TIME = TIME.V - COMM.TIME
N.SURVIVOR = REAL.F(XX)

''TO PRINT INTERMEDIATE RESULTS
SKIP 2 LINES
PRINT 5 LINES WITH Z,N.LANE,GO,MEAN.SURVIVOR,
VAR.SURVIVOR,MEAN.BATTLE.TIME,VAR.BATTLE.TIME THUS
FOR RUN **
LANE *          MEAN      VARIANCE
ARTY ***         ----       --------
NUMBER SURVIVORS  ***.*        ***.*
BATTLE TIME       ***.*        ***.*

FOR EACH MINE OF MINE.FIELD DO
  REMOVE MINE FROM MINE.FIELD
  DESTROY MINE
LOOP

 SKIP 1 LINE
 PRINT 1 LINE WITH N.BLUE.ROUND THUS
 ROUNDS FIRED BY BLUE ****

 SKIP 1 LINE
 PRINT 1 LINE WITH N.RED.ROUND THUS
 ROUNDS FIRED BY RED ****

''TO RESET/RESTART SIMULATION FOR NEXT RUN
IF GRAPHICS = "YES"
 FOR I = 1 TO N.BLUE.ROUND DO
   DESTROY XPLODE CALLED XB(I)
 LOOP
 N.BLUE.ROUND = 0

 FOR I = 1 TO N.RED.ROUND DO
```

```
      DESTROY XPLODE CALLED XR(I)
    LOOP
    N.RED.ROUND = 0

    FOR I = 1 TO N DO
      IF ELEMENT.STATUS(V(I)) EQ "OBSTACLE"
        DESTROY  DEAD.TANK CALLED DT(I)
      ALWAYS
    LOOP
  ALWAYS
LOOP END
```

C. ROUTINE LAY.MINE.FIELD

```
ROUTINE LAY.MINE.FIELD  ''THIS ROUTINE LAYS THE MINEFIELD

  DEFINE INTERVAL, ''EQUAL SEGMENTS OF MINEFIELD WIDTH
  MEAN,            ''CENTER OF INTERVAL
  SD               ''DISTRIBUTION STANDARD DEVIATION
  AS REAL VARIABLES
  DEFINE SEED1 TO MEAN INT.F(RANDOM.F(2)*10)

  ''TO AVOID 0 SEED VALUE
  SEED = SEED1
  IF SEED = 0
   SEED = 1
  ALWAYS

  FOR EACH MINE.BELT DO

    INTERVAL = MF.WIDTH/N.MINES
    DENSITY(MINE.BELT) = 1/INTERVAL

    SKIP 2 LINES
```

```
PRINT 2 LINES WITH MINE.BELT, DENSITY USING UNIT 2 THUS
MINE DENSITY OF BELT * : *.*** MINES PER METER
-------------------------------------------
PRINT 2 LINES USING UNIT 2 THUS
MINE #     EASTING    NORTHING
------     -------    --------

FOR I = 1 TO N.MINES(MINE.BELT) DO

  CREATE  A MINE
  MINE.NUMBER = (MINE.BELT*10000) + I
  MEAN = ((I*2-1)*INTERVAL)/2
  SD = INTERVAL/6

  ''NORMAL SAMPLING TO DETERMINE MINE LOCATION
  MINE.X = NORMAL.F(MEAN,SD,SEED)
  MINE.Y = NORMAL.F(DEPTH(MINE.BELT),MINE.DEV,2)

  IF MINE.X LE 0
   MINE.X = 0
  ALWAYS

  PRINT 1 LINE WITH MINE.NUMBER, GR.ZONE,
  MINE.X + GR.BASE.E,MINE.Y + GR.BASE.N USING UNIT 2 THUS
*****     ** *****.*   *****.*

  IF MINE.X GT 0 OR MINE.Y GT 0
   FILE THE MINE IN THE MINE.FIELD
   INPLACED = 1
   MINE.STATUS = "ACTIVE"
   RADIUS = TEMP.RADIUS
   ''MINE TYPES ARE HOMOGENEOUS WITHIN BELTS
   MINE.TYPE = MINE.B.TYPE(MINE.BELT)
   IF GRAPHICS EQ "YES"
    LET LOCATION.A(MINE) = LOCATION.F(MINE.X(MINE),
```

```
                                        MINE.Y(MINE))
              SHOW MINE WITH "MINE.ICN"
            ALWAYS
          ELSE
            DESTROY MINE
          ALWAYS

        LOOP

      LOOP

      SKIP 2 LINES
      PRINT 2 LINES WITH TOT.ACTIVE USING UNIT 2 THUS
      MINEFIELD CONTAINS **** MINES
      ------------------------------

      INITIAL = TOT.ACTIVE


RETURN    END
```

D. PROCESS BLUE.ARTY.ATK.MINE.FIELD

```
 PROCESS BLUE.ARTY.ATK.MINE.FIELD
 ''THIS PROCES FIRES BLUE ARTILLERY AGAINST THE MINEFIELD

    DEFINE
    IMPACT.X,  ''ARTILLERY ROUND IMPACT EASTING
    IMPACT.Y,  ''            "       NORTHING
    DISTANCE   ''DISTANCE FORM MINE TO IMPACT
    AS REAL VARIABLES

    DEFINE SEED1 TO MEAN INT.F(RANDOM.F(3)*10)
```

```
  IF SEED = 0
   SEED = 1
  ALWAYS

  IF GO EQ "NO" OR GO EQ "no"
   RETURN
  ALWAYS

''TO DETERMINE THE NUMBER OF AIMPOINTS
  FOR EACH WEAPON DO
    IF WEAPON.UNIT = "BLUE"
      N.AIMPOINT = INT.F(MAX.DISTANCE/(2*LETHAL.RADIUS))
    ALWAYS
  LOOP

''TO DETERMINE DISTANCE AND ASSESS DAMAGE TO MINEFIELD
    FOR EACH LANE DO
    FOR EACH WEAPON DO
    IF WEAPON.UNIT = "BLUE"
    FOR K = 1 TO N.AIMPOINT DO
      FOR I = 1 TO N.VOLLEYS DO

        ''NORMAL SAMPLING FOR IMPACT LOCATION
          IMPACT.X = NORMAL.F(START.X,ERROR.X,SEED)
          IMPACT.Y = NORMAL.F((2*K-1)*LETHAL.RADIUS,
                                   ERROR.Y,SEED)
          ADD 1 TO N.BLUE.ROUND ''TO TRACK EXPENDITURE
          IF GRAPHICS EQ "YES"
            CREATE AN XPLODE CALLED XB(N.BLUE.ROUND)
        DISPLAY XB(N.BLUE.ROUND) WITH "XPLODE.ICN" AT
                                (IMPACT.X,IMPACT.Y)
          ALWAYS
''TO SORT (FILTER) MINES FOR DISTANCE
          FOR EACH MINE OF MINE.FIELD WITH
          ((IMPACT.X-MINE.X) +  (IMPACT.Y-MINE.Y)) LE
```

```
                    (1.42*LETHAL.RADIUS) DO
          DISTANCE = SQRT.F((IMPACT.X-MINE.X)**2 +
                    (IMPACT.Y-MINE.Y)**2)
''DAMAGE ASSESSMENT AGAINST MINE
          IF DISTANCE LE LETHAL.RADIUS/2
           PRINT 1 LINE WITH WEAPON.UNIT,
           WEAPON.TYPE USING UNIT 2 THUS
  MINE DESTROYED BY **** ******** AT GRID:

PRINT 1 LINE WITH MINE.NUMBER,GR.ZONE,MINE.X+GR.BASE.E,
MINE.Y+GR.BASE.N USING UNIT 2 THUS
   *****  ** *****.*  *****.*

           SUBTRACT 1 FROM TOT.ACTIVE
           MINE.STATUS = "DESTROYED BY BLUE ARTY"
           IF GRAPHICS = "YES"
            DCOLOR.A(ICON.A(MINE)) = 6
            DISPLAY MINE
           ALWAYS
          ALWAYS
         LOOP
       LOOP
     LOOP
     ALWAYS
     LOOP
     LOOP
''TO MAINTAIN ACCOUNTABILITY OF MINES
  PRINT 2 LINES WITH INITIAL - TOT.ACTIVE, TOT.ACTIVE USING
  UNIT 2 THUS MINES DESTROYED **** MINES REMAINING ****

END
```

E. ROUTINE MAKE.ROUTE

```
ROUTINE MAKE.ROUTE

''THIS ROUTINE IS DESIGNED TO GENERATE A ROUTE FOR EACH ELE
''A 3-DIMENSIONAL ARRAY IS INITIALIZED FOR THE DATA
''FOR EACH ELE. THE ARRAY IS LABELED 'MOVEMENT.PLAN' AND IS
''OF SIZE 'NUMBER OF ELEMENT x NUMBER OF CHECKPOINTS x 5

 DEFINE I AS INTEGER VARIABLES
 DEFINE J AS AN INTEGER VARIABLE
 DEFINE K AS AN INTEGER VARIABLE
 DEFINE L AS AN INTEGER VARIABLE
 DEFINE SEED1 TO MEAN INT.F(RANDOM.F(5)*10)

''TO AVOID 0 SEED
SEED = SEED1
IF SEED = 0
  SEED = 1
ALWAYS


''TO DETERMINE NAVIGATION PLAN FOR EACH BREACHER OF EACH
LANE FOR EACH LANE WITH LANE GE 2 DO
  AAA = 0

''TO TALLY BREACHERS ON EACH LANE
  FOR I = 1 TO LANE-1 DO
    AAA = AAA + N.ELEMENT(LANE-1)
  LOOP

  FOR I = (AAA + 1) TO (N.ELEMENT + AAA) DO  ''FOR EACH ELEMENT
     CUR.Y.LOC = (AAA-1)*50      '' RESET CHECKPOINT Y LOCATION

     FOR L = 1 TO NUM.CP DO     ''AND THEN FOR EACH CHECKPOINT
                             ''GENERATE, USING A NORMAL (0,1)
```

```
        '' A NAVIGATION ERROR IN THE X DIMENSION
MOVEMENT.PLAN(I,L,1) = NORMAL.F(START.X,2.0,SEED)
        ''AND ASSIGN A Y LOCATION BASED ON CHECKPOINT
            ''SEQUENCE AND INTERVAL
MOVEMENT.PLAN(I,L,2) = CUR.Y.LOC
CP(I,L) = CUR.Y.LOC
      ''AND INCREMENT CHECKPOINT Y-LOCATION COUNTER

CUR.Y.LOC = CUR.Y.LOC + 20.0

 LOOP
LOOP
for K = N.ELEMENT(LANE-1)+1 to N.ELEMENT(LANE-1)+
                          N.ELEMENT(LANE) DO

 J = NUM.CP - 1

 MOVEMENT.PLAN(K,1,5) = 1

 FOR I = 1 TO J DO   ''FOR EACH CHECKPOINT, EXCEPT THE LAST

      ''CALCULATE SLOPE OF LINE CONNECTING CHECKPOINT
      ''WITH NEXT CHECKPOINT AND STORE RESULT
      ''IN ARRAY
  MOVEMENT.PLAN(K,I,3) =
  ((MOVEMENT.PLAN(K,I+1,2) - MOVEMENT.PLAN(K,I,2)) /
  (MOVEMENT.PLAN(K,I+1,1) - MOVEMENT.PLAN(K,I,1)))

        ''CALCULATE Y-INTERCEPT OF LINE CONNECTING
        ''CURRENT CHECKPOINT WITH NEXT CHECKPOINT
        ''AND STORE RESULT IN ARRAY
  MOVEMENT.PLAN(K,I,4) =
  MOVEMENT.PLAN(K,I,2) - (MOVEMENT.PLAN(K,I,3) *
  MOVEMENT.PLAN(K,I,1))
```

```
  LOOP

LOOP LOOP FOR EACH LANE WITH LANE EQ 1 DO
FOR I = 1 TO N.ELEMENT DO  ''FOR EACH ELEMENT
  CUR.Y.LOC = (-1)*50  ''RESET CHECKPOINT Y LOCATION COUNTER

   FOR L = 1 TO NUM.CP  DO ''AND THEN FOR EACH CHECKPOINT

         ''GENERATE, USING A NORMAL (0,1) DISTRIBUTION,
         ''A NAVIGATION ERROR IN THE X DIMENSION
    MOVEMENT.PLAN(I,L,1) = NORMAL.F(START.X,2.0,SEED)
         ''AND ASSIGN A Y LOCATION BASED ON CHECKPOINT
                  ''SEQUENCE AND INTERVAL
    MOVEMENT.PLAN(I,L,2) = CUR.Y.LOC
    CP(I,L) = CUR.Y.LOC
       ''AND INCREMENT CHECKPOINT Y-LOCATION COUNTER
    CUR.Y.LOC = CUR.Y.LOC + 20.0

  LOOP

LOOP
FOR K = 1 TO TOT.ELEMENT DO

  J = NUM.CP - 1

  MOVEMENT.PLAN(K,1,5) = 1

  FOR I = 1 TO J DO ''FOR EACH CHECKPOINT, EXCEPT THE LAST

             ''CALCULATE SLOPE OF LINE CONNECTING CHECK
             ''WITH NEXT CHECKPOINT AND STORE RESULT
             ''IN ARRAY
   MOVEMENT.PLAN(K,I,3) =
   ((MOVEMENT.PLAN(K,I+1,2) - MOVEMENT.PLAN(K,I,2)) /
   (MOVEMENT.PLAN(K,I+1,1) - MOVEMENT.PLAN(K,I,1)))
```

```
            ''CALCULATE Y-INTERCEPT OF LINE CONNECTING
            ''CURRENT CHECKPOINT WITH NEXT CHECKPOINT
            ''AND STORE RESULT IN ARRAY
     MOVEMENT.PLAN(K,I,4) =
    MOVEMENT.PLAN(K,I,2) - (MOVEMENT.PLAN(K,I,3) *
    MOVEMENT.PLAN(K,I,1))

   LOOP

  LOOP LOOP RETURN


END ''MAKE.ROUTE
```

F. ROUTINE NEXT.ENCOUNTER

```
ROUTINE NEXT.ENCOUNTER GIVEN ALLEY,ID
''THIS ROUTINE DETERMINE THE NEXT ENCOUNTER FOR EACH
BREACHER
''EITHER OBSTACLE, MINE, OR CHECKPOINT

  DEFINE
  NUMBER,ID
  AS INTEGER VARIABLES
  DEFINE
  CP.DISTANCE, ''DISTANCE TO CHECK POINT
  M.DISTANCE,  ''DISTANCE TO MINE
  DURATION,    ''DISTANCE/SPEED
  X1,X2,       ''BREACHER EASTING, NORTHING
  SCH.TIME,    ''SCHEDULE FOR AN EVENT
  OBS.DISTANCE ''DISTANCE TO OBSTACLE
  AS REAL VARIABLES
```

```
''TO STOP PROGRAM IF NO BREACHERS
 IF TOT.ELEMENT LE 0
   STOP
 ALWAYS
''TO EXIT IF BREACHER THROUGH MINEFIED OR BECOMES DISABLED
 IF ELEMENT.STATUS(V(ID)) EQ "THROUGH" OR ELEMENT.STATUS(V(ID))
                                   EQ "OBSTACLE"
   RETURN
 ALWAYS

  CALL DISTANCE.TO.CP GIVING ID YIELDING CP.DISTANCE, X1, X2
  CALL DISTANCE.TO.MINE GIVING ID YIELDING NUMBER, M.DISTANCE
  CALL DISTANCE.TO.OBS GIVING ID YIELDING OBS.DISTANCE

''PRINT 1 LINE WITH ID, CP.DISTANCE, M.DISTANCE, OBS.DISTANCE
''THUS FROM NEXT.ENCOUNTER ELEMENT **. CP AT ***.** ''MINE AT
****.** OB AT *****.**

  IF CP.DISTANCE < MIN.F(M.DISTANCE,OBS.DISTANCE)
    ''SCHEDULE CP ENCOUNTER
    CALL DELTA.TIME GIVING ALLEY,ID,CP.DISTANCE YIELDING DURA-
TION
    SCH.TIME = TIME.V + DURATION

''PRINT 2 LINES WITH ID, SCH.TIME, CP.DISTANCE USING UNIT 2
''THUS ELEMENT **'S NEXT MOVEMENT ENCOUNTER IS A CP AT
''TIME ***.** THE CP IS ***.** METERS AWAY

    SCHEDULE A NEW.CP GIVING ALLEY,ID AT SCH.TIME
  ENDIF

  IF M.DISTANCE < MIN.F(CP.DISTANCE,OBS.DISTANCE)
    ''SCHEDULE MINE ENCOUNTER
    CALL DELTA.TIME GIVING ALLEY,ID,M.DISTANCE YIELDING DURA-
TION
```

```
    SCH.TIME = time.v + DURATION

  ''PRINT 1 LINE WITH ID, NUMBER, SCH.TIME USING UNIT 2 THUS
 ''ELEMENT ** S NEXT MOVEMENT ENCOUNTER IS MINE ***** AT ***.**

    SCHEDULE A MINE.ENCOUNTER GIVING ALLEY,ID,NUMBER AT
SCH.TIME

  ''   PRINT 1 LINE WITH ID, NUMBER, SCH.TIME USING UNIT 2 THUS
  ''   ELEMENT ** WILL HIT MINE ***** AT TIME ****.**

  ENDIF

  IF OBS.DISTANCE < MIN.F(M.DISTANCE,CP.DISTANCE)
    ''SCHEDULE OBSTACLE ENCOUNTER
    CALL DELTA.TIME GIVING ALLEY,ID,OBS.DISTANCE
                                YIELDING DURATION
    SCH.TIME = TIME.V + DURATION

    SCHEDULE AN OBSTACLE.ENCOUNTER GIVING ALLEY,ID AT
SCH.TIME

   '' PRINT 1 LINE WITH ID, SCH.TIME USING UNIT 2 THUS
   '' ELEMENT ** WILL HIT AN OBSTACLE AT TIME ****.**

  ENDIF
  RETURN
  END
```

G. ROUTINE DISTANCE.TO.CP

```
ROUTINE DISTANCE.TO.CP GIVEN ID YIELDING DISTANCE, DEL.X, DEL.Y
```

```
''THIS ROUTINE DETERMINES THE DISTANCE BETWEEN THE IDENTIFIED
''ELEMENT AND THE NEXT CHECKPOINT ON THAT ELEMENTS
''MOVEMENT PLAN.

DEFINE ID AS AN INTEGER VARIABLE       ''ELEMENT,LANE INDEX
DEFINE CURR.CP AS AN INTEGER VARIABLE  ''CHECKPOINT INDEX
DEFINE DISTANCE AS A REAL VARIABLE ''DIST BETWEEN ELE & CP
DEFINE DEL.X AS A REAL VARIABLE        ''DIFFERENCE IN X
DEFINE DEL.Y AS A REAL VARIABLE        ''DIFFERENCE IN Y

                  ''DETERMINE ELEMENT'S CURRENT LOCATION
                  ''ON ITS RESPECTIVE MOVEMENT PLAN, THEN
IF ELEMENT.STATUS(V(ID))EQ"OBSTACLE" OR
                  ELEMENT.STATUS(V(ID)) EQ "THROUGH"
 RETURN
ALWAYS


CALL FIND.CURRENT.CP GIVING ID YIELDING CURR.CP

IF ELEMENT.STATUS(V(ID)) EQ "ACTIVE"
                  ''CALCULATE THE X DISTANCE BETWEEN THEN
                  ''NEXT CHECKPOINT AND THE ELEMENT, THEN
 DEL.X = MOVEMENT.PLAN(ID,CURR.CP + 1,1) - ELEMENT.X(V(ID))

                  ''CALCULATE THE Y DISTANCE BETWEEN THE
                  ''NEXT CHECKPOINT AND THE ELEMENT, THEN
 DEL.Y = MOVEMENT.PLAN(ID,CURR.CP + 1,2) - ELEMENT.Y(V(ID))

                ''USE THE PYTHAGORIUM THEOREM TO DETERMINE
                ''THE STRAIGHT LINE DISTANCE BETWEEN THE
                ''ELEMENT AND THE NEXT CHECKPOINT
ALWAYS
IF ELEMENT.STATUS(V(ID)) EQ "BYPASS"
 DEL.X = BYPASS.MAP(ID,CURR.CP+1,1) - ELEMENT.X(V(ID))
```

```
DEL.Y = BYPASS.MAP(ID,CURR.CP+1,2) - ELEMENT.Y(V(ID))
ALWAYS

DISTANCE = SQRT.F(DEL.X**2 + DEL.Y**2) IF DISTANCE LE 0
 DISTANCE = 1.0 ALWAYS
                    ''RETURN THE DISTANCES RETURN END
```

H. ROUTINE DISTANCE.TO.MINE

```
ROUTINE DISTANCE.TO.MINE GIVEN ID YIELDING MINE.ID,
                         RANGE.TO.MINE

''THIS ROUTINE DETERMINES THE DISTANCE THAT A MINE LIES FROM
''THE PATH OF A SPECIFIED ELEMENT. INPUTS ARE THE ELEMENT ID.
''IF THE DISTANCE IS LESS THAN THE RADIUS OF THE MINE,
''AN ENCOUNTER RWILL OCCUR.

DEFINE
ID,             ''BREACHER NUMBER
MINE.ID,        ''MINE NUMBER
POSSIBLE.MINE,  ''CANDIDATE MINE FOR ENCOUNTER
CURR.CP         ''BREACHER'S CURRENT CHECK POINT
AS INTEGER VARIABLES

DEFINE
SLOPE,
INTERCEPT,
MINE.SLOPE,
Y.BOTTOM,
Y.TOP,
X.LEFT,
X.RIGHT,
Y2,Y3,
```

```
DISTANCE.TO.MINE.ENC,
POSSIBLE.RANGE,
MISS.DIS,        " MISS DISTANCE
X1,X2,X3,Y1,
RANGE.TO.MINE
AS REAL VARIABLES

RANGE.TO.MINE = 1000.0


CALL FIND.CURRENT.CP GIVING ID YIELDING CURR.CP

"RECORD APPROPRIATE VALUES
 SLOPE = MOVEMENT.PLAN(ID,CURR.CP,3)
 INTERCEPT = MOVEMENT.PLAN(ID,CURR.CP,4)

IF ELEMENT.STATUS(V(ID)) = "BYPASS"
 SLOPE = BYPASS.MAP(ID,CURR.CP,3)
 INTERCEPT = BYPASS.MAP(ID,CURR.CP,4)
ALWAYS
"SET FILTERS
                "FILTERS ARE USED TO ELIMINATE
                "FROM INSPECTION THOSE MINES THAT
                "ARE TOO FAR AWAY TO BE POSSIBLE
                "ENCOUNTERS. THIS REDUCES THE NUMBER
                "OF CALCULATIONS REQUIRED
 Y.BOTTOM = ELEMENT.Y(V(ID))
 Y.TOP = ELEMENT.Y(V(ID)) + 24.0 "CAREFULL.... DO NOT SET UPPER
                  "SO LOW AS TO PERMIT UNINTENTIONAL
                   "MOVEMENT INTO UNCOMPUTED AREAS

 IF              (MOVEMENT.PLAN(ID,CURR.CP,1)               <
MOVEMENT.PLAN(ID,CURR.CP+1,1))
   X.LEFT = MOVEMENT.PLAN(ID,CURR.CP,1) - 12.0
   X.RIGHT = MOVEMENT.PLAN(ID,CURR.CP+1,1) +12.0
```

```
ELSE
  X.LEFT = MOVEMENT.PLAN(ID,CURR.CP+1,1) - 12.0
  X.RIGHT = MOVEMENT.PLAN(ID,CURR.CP,1) + 12.0
ALWAYS
                    ''FOR BYPASSING ELEMENTS
IF ELEMENT.STATUS(V(ID)) = "BYPASS"
 IF BYPASS.MAP(ID,CURR.CP,1) LE BYPASS.MAP(ID,CURR.CP+1,1)
   X.LEFT = BYPASS.MAP(ID,CURR.CP,1) - 12.0
   X.RIGHT = BYPASS.MAP(ID,CURR.CP+1,1) + 12.0
 ELSE
   X.LEFT = BYPASS.MAP(ID,CURR.CP+1,1) - 12.0
   X.RIGHT = BYPASS.MAP(ID,CURR.CP,1) + 12.0
 ALWAYS
ALWAYS

''USE FILTERS TO REDUCE NUMBER OF CANDIDATE MINES
                ''LOOK AT EACH MINE
FOR EACH MINE IN MINE.FIELD WITH MINE.STATUS = "ACTIVE" DO
  IF MINE.Y > Y.BOTTOM AND MINE.Y < Y.TOP AND MINE.X > X.LEFT
                              AND MINE.X < X.RIGHT

''PRINT 1 LINE WITH MINE.NUMBER, MINE.X, MINE.Y USING UNIT 2
''THUS MINE ***** IS A CANDIDATE. ITS COORDINATES ARE:
''***.**,***.**

                  ''CALCULATE SLOPE OF LINE PERPINDICULAR
                     ''TO PATH EQUATION USING M1*M2=-1
     MINE.SLOPE = -1.0 / SLOPE
                     ''USE POINT-SLOPE FORMULA,
                  ''Y-Y1=M(X-X1) TO DEVELOP EQUATION FOR
                  ''MINE-LINE, COMBINE WITH EQUATION FOR
                     ''MOVEMENT PATH LINE TO SOLVE FOR
                     ''INTERSECTION POINT
     X1 = (MINE.Y - INTERCEPT - (MINE.SLOPE * MINE.X)) /
                  (SLOPE - MINE.SLOPE)
```

```
        Y1 = X1 * SLOPE + INTERCEPT

''      PRINT 1 LINES WITH X1, Y1 USING UNIT 2 THUS
''      MINE INTERCEPT POINT IS ***.**, ***.**

        ''COMPUTE MINE DISTANCE FROM MOVEMENT PATH
        X2 = X1 - MINE.X
        Y2 = Y1 - MINE.Y
        MISS.DIS = SQRT.F(X2**2 + Y2**2) - WIDTH(V(ID))/2
                  ''COMPUTE MINE ENCOUNTER DISTANCE
                  ''FROM MOVING ELEMENT
        X3 = X1 - ELEMENT.X(V(ID))
        Y3 = Y1 - ELEMENT.Y(V(ID))
        DISTANCE.TO.MINE.ENC = SQRT.F(X3**2 + Y3**2)

''PRINT 1 LINE WITH MINE.NUMBER, MISS.DIS USING UNIT 2 THUS
''        MINE ***** WAS ***.*** FROM PATH.

''LIST MISS.DIS
        IF MISS.DIS LT (RADIUS(MINE) + (.5*WIDTH(V(ID))))
         POSSIBLE.MINE = MINE.NUMBER
         POSSIBLE.RANGE = DISTANCE.TO.MINE.ENC

''         PRINT 1 LINE WITH MINE.NUMBER THUS
''         !!!!LOOKS LIKE MINE NUMBER ***** WILL GET HIT

          IF POSSIBLE.RANGE < RANGE.TO.MINE
           MINE.ID = POSSIBLE.MINE
           RANGE.TO.MINE = POSSIBLE.RANGE
          ALWAYS
        ALWAYS
    ALWAYS
 LOOP

''PRINT 1 LINE WITH MINE.ID,RANGE.TO.MINE USING UNIT 2 THUS
```

```
''     MINE ***** WILL BE ENCOUNTERED IN ***.** METERS

RETURN
END ''DISTANCE TO MINE
```

I. FUNCTION DISTANCE.TO.OBS

```
FUNCTION DISTANCE.TO.OBS GIVEN ID YIELDING  DISTANCE

''THIS FUNCTION DETERMINES THE DISTANCE FROM THE GIVEN
''ELEMENT TO ANY OBASTCLES WITHIN THE ELEMENTS CURRENT
''CHECKPOINT SECTOR. THE OBSTACLE ID AND THE DISTANCE TO THE
''OBSTACLE'S EDGE THAT INTERSECTS THE ELEMENTS PATH ARE
''RETURNED

DEFINE ID AS AN INTEGER VARIABLE              ''ELEMENT ID
DEFINE OBS.ID AS AN INTEGER VARIABLE           ''OBSTACLE ID
DEFINE DISTANCE AS A REAL VARIABLE  '' DISTANCE TO OBS ID'ED
DEFINE LITTLE.D AS A REAL VARIABLE
DEFINE BIG.D AS A REAL VARIABLE
DEFINE FRACTION AS A REAL VARIABLE
DEFINE CURR.CP AS AN INTEGER VARIABLE ''CURRENT ELE CHECKPNT
DEFINE SLOPE AND Y.INT AS REAL VARIABLES ''FOR ELE PATH EQN
DEFINE K AS AN INTEGER VARIABLE               ''COUNTER
DEFINE OBS.SLOPE AS A REAL VARIABLE ''SLOPE OF PERPENDICULAR
DEFINE E.X AND E.Y AS REAL VARIABLES
DEFINE OB.X AND OB.Y AS REAL VARIABLES
DEFINE X1 AND Y1 AS REAL VARIABLES    ''LOC OF INTERCEPT
DEFINE X2 AS A REAL VARIABLES        ''DELTA DISTANCE, EACH AXIS
DEFINE X3 AND Y3 AS REAL VARIABLES
DEFINE C.X AND C.Y AS REAL VARIABLES
DEFINE OFFSET AS A REAL VARIABLE      ''STRAIGHT LINE DISTANCE
DEFINE DIS.TO.OBS AS A REAL VARIABLE  ''DISTANCE TO OBSTACLE
```

```
DEFINE OB.RAD AS A REAL VARIABLE      ''RADIUS OF OBSTACLE
DEFINE E.RAD AS A REAL VARIABLE
DEFINE CONTACT.RAD AS A REAL VARIABLE
DEFINE DEL.X, DEL.Y AS REAL VARIABLES
DEFINE F.DEL.X, F.DEL.Y AS REAL VARIABLES ''CH FR ELE TO OBS
DEFINE CURRENT.BEST AS A REAL VARIABLE
DEFINE CHECKED.LIST AS A 1-DIMENSIONAL INTEGER ARRAY

        ''INITIALIZE DISTANCE AND ID WITH VALUES WHICH WILL
        ''NOT POSSIBLY BE SELECTED AS THE NEXT ENCOUNTER TO
        ''PREVENT A NULL SOLUTION (ALSO PREVENT NULL
            ''SUBSCRIPT FOR OBSTACLE ARRAY)

IF ELEMENT.STATUS(V(ID)) = "OBSTACLE"
 RETURN
ALWAYS

DISTANCE = 10000.0
OBS.ID = 1
E.X = ELEMENT.X(V(ID))
E.Y = ELEMENT.Y(V(ID))
E.RAD = ELEM.RADIUS(V(ID))
RESERVE CHECKED.LIST(*) AS NUM.CP/2

             ''STATUS OF 2 IS USED TO INDICATE ELEMENT IS
             ''BYPASSING - SINCE THIS PROGRAM IS NOT
             ''DESIGNED TO ALLOW MORE THAN ONE ENCOUNTER
             ''EVENT PER ELEMENT AT A TIME, IF THE ELEMENT
             ''IS ALREADY BYPASSING, THIS ROUTINE IS NOT
                   ''ALLOWED TO RUN.
IF ELEMENT.STATUS(V(ID)) NE "BYPASS"

 CALL FIND.CURRENT.CP GIVING ID YIELDING CURR.CP
```

```
''++++++++++++++++DEBUG MESSAGE - USED TO SHOW CP
USED
''PRINT 1 LINE WITH ID, CURR.CP THUS
''ELEMENT ** IS CHECKING DISTANCE TO OBSTACLE USING CP OF **

    ''  FIRST NEED TO FIND OUT IF OUR SCHEDULED PATH,
    ''ADJUSTED FOR ELEMENT WIDTH, WILL INTERSECT WITH AN
    ''EXISTING OBSTACLE. OBSTACLE DATA CONSISTS OF THE
    ''COORDINATES OF THE CENTER OF THE OBSTACLE AND THE
    ''RADIUS OF THE OBSTACLE. BOTH VALUES ARE DYNAMIC, WITH
    ''THE COORDINATES CHANGING TO INDICATE THE CENTROID OF
    ''A MULTIPLE ELEMENT OBSTACLE, AND THE RADIUS INCREASING
    ''TO GIVE THE RADIUS OF THE LARGER OBSTACLE.
    ''  THIS ROUTINE CHECKS BOTH THE CURRENT AND THE
    ''NEXT CP INTERVAL - THIS ASSUMES THAT THE OBSTACLE
    ''WILL NOT HAVE A RADIUS GREATER THAN 1.5 CHECKPOINT
    ''INTERVALS. IF OBSTACLES ARE TO BE ALLOWED GREATER SIZE,
    ''THEN ADDITIONAL INTERVALS WILL NEED TO BE CHECKED.

    ''  DETERMINE IF AN OBSTACLE HAS EXISTS IN THE CURRENT
    ''CHECKPOINT INTERVAL;
                        ''LINE EQUATION PARAMETERS FOR
                        ''ELEMENTS CURRENT PATH
 SLOPE = MOVEMENT.PLAN(ID,CURR.CP,3)  ''SLOPE OF PATH LINE
 Y.INT = MOVEMENT.PLAN(ID,CURR.CP,4)  ''Y-INTERCEPT OF PATH LINE

FOR I = 1 TO NUM.CP/2 DO
 CHECKED.LIST(I) = 0
LOOP

'START.SEARCH'

  CURRENT.BEST = 1000.0
  K = 0
```

```
for EACH ELEMENT OF OBSTACLE.LIST WITH ELEMENT.Y GE E.Y DO
  IF (ELEM.RADIUS > 0),      ''OBSTACLE EXISTS
    IF(CHECKED.LIST(ELEMENT.NUM) NE 1) ''AND NOT REJECTED
      if ELEMENT.Y < CURRENT.BEST
         CURRENT.BEST = ELEMENT.Y
         K = ELEMENT.NUM
         LEAVE
      ENDIF
    ENDIF
  ENDIF
LOOP

IF (K = 0)
''      PRINT 1 LINE WITH ID THUS
''        NO OBSTACLE FOUND IN ELEMENT ** S PATH
   RETURN
ENDIF

OB.RAD = ELEM.RADIUS(V(K))
CONTACT.RAD = OB.RAD + E.RAD
OB.X = ELEMENT.X(V(K))
OB.Y = ELEMENT.Y(V(K))

''+ + + + + + + + + + + + + + + + + + DEBUG MESSAGE -- IDENTIFY OBSTACLE
''PRINT 1 LINE WITH ID, K THUS
''EXAMINING RELATIONSHIP BETWEEN ELE ** & OBS **
''PRINT 1 LINE WITH E.X, E.Y, OB.X, OB.Y THUS
''ELEMENT IS AT X = ***.**, Y = ***.**  OBSTACLE AT
''X = ***.**, Y = ***.**

                 ''DETERMINE INTERCEPT POINT OF THE MOVEMENT
                 ''PATH AND THE PERPENDICULAR LINE THAT PASSES
                 ''THROUGH THE CENTER OF THE OBSTACLE
OBS.SLOPE = -1.0 / SLOPE
```

```
            ''CALCULATE COORDINATES OF INTERCEPT POINT
            ''USING POINT-SLOPE FORMULA
X1 = (OB.Y - Y.INT - (OBS.SLOPE * OB.X)) / (SLOPE - OBS.SLOPE)
Y1 = X1 * SLOPE + Y.INT

              ''COMPUTE OBSTACLE CENTER DISTANCE FROM
              ''MOVEMENT PATH FOR EACH AXIS
X2 = X1 - OB.X

              ''DETERMINE THE STRAIGHT LINE DISTANCE
              ''BETWEEN THE INTERSECTION POINT AND THE
           ''CENTER OF THE OBSTACLE (PERPENDICULAR LINE)
OFFSET = ABS.F(X2/2)

        ''IN ORDER TO ALLOW FOR
        ''OBSTACLES TO OVERLAP CHECKPOINT LINES,
        ''OBSTACLES THAT ARE TOO FAR AWAY, AND WHICH WILL
        ''NOT BE ENCOUNTED USING THE CURRENT PATH EQUATION
        ''MAY BE EXAMINED. IF THIS HAPPENS, THE OFFSET
       ''VALUE CALCULATED WILL BE GREATER THAT THE COMBINED
       ''ELEMENT-OBSTACLE RADIUS. IF THIS HAPPENS, DISCARD
       ''OBSTACLE AS A CANDIDATE.
              ''I.E. DETERMINE IF CONTACT TAKES PLACE
''  LIST OFFSET,CONTACT.RAD
IF (OFFSET > CONTACT.RAD)

''++++++++++++++++++++DEBUG MESSAGE
''PRINT 1 LINE WITH ID, CURR.CP, K THUS
''CANCELLED BYPASS CALC FOR ELE **, AT CP **, FOR OBS **

  CHECKED.LIST(K) = 1  ''SELECTION WAS NO GOOD
  GO TO 'START.SEARCH'
ENDIF

DEL.X = X1 - E.X  ''CHANGE IN X FROM ELEMENT TO INTERSECTION
```

```
  DEL.Y = YI - E.Y  ''CHANGE IN Y FROM ELEMENT TO INTERSECTION
              ''MISSING SIDE OF RIGHT TRIANGLE
  LITTLE.D = SQRT.F(CONTACT.RAD**2 - OFFSET**2)
              ''DISTANCE FROM ELEMENT TO INTERSECTION
  BIG.D = SQRT.F(DEL.X**2 + DEL.Y**2)
          ''PERCENTAGE OF THAT DISTANCE COVERED BY OBSTACLE
  FRACTION = LITTLE.D / BIG.D
              ''EXPRESSED AS A CHANGE IN X
  X3 = FRACTION * DEL.X
              ''AND AS A CHANGE IN Y
  Y3 = FRACTION * DEL.Y

              ''BASED ON PATH SLOPE, DETERMINE IF CHANGE IN X
              ''IS POSITIVE OR NEGATIVE
  IF SLOPE > 0
   C.X = X1 - X3
  ENDIF

  IF SLOPE < 0
   C.X = X1 + X3
  ENDIF
              ''AND DETERMINE CHANGE IN Y. RESULTING C.X, C.Y
              ''ARE THE COORDINATES OF THE FIRST OBSTACLE RADIUS
              '' - PATH LINE INTERSECTION. THE C.X CAN BE USED
              ''TO DETERMINE WHICH DIRECTION TO BYPASS (SHORTEST
              ''DISTANCE AROUND).
  C.Y = Y1 - Y3
  IF C.Y < E.Y
''PRINT 1 LINE WITH ID, K THUS
''ELEMENT ** WAS ALREADY BEYOND BEGINNING OF OBS ** - RETRY
   CHECKED.LIST(K) = 1
   GO TO 'START.SEARCH'
  ENDIF
''+ + + + + + + +DEBUG MESSAGE  CHECK ON TURN DIRECTION LOGIC
''PRINT 1 LINE WITH ID,C.Y, E.Y THUS
```

```
''ELEMENT ** HAS A C.Y OF ***.** AND AN E.Y OF ***.**

            ''CALL BYPASS TO DETERMINE LOCATION OF 'EXIT'
            ''FROM PATH TO BYPASS THIS OBSTACLE, K IS THE
            ''OBSTACLE NUMBER. RESULT IS THE CHANGE IN X
            ''AND THE CHANGE IN Y FROM THE ELEMENTS CURRENT
            ''POSITION TO THE EXIT POINT
  CALL BYPASS GIVING ID, K, C.X  YIELDING F.DEL.X, F.DEL.Y

  F.DEL.X = F.DEL.X - E.X ''CHG IN X FROM ELEMENT TO EXIT POINT
  F.DEL.Y = F.DEL.Y - E.Y ''CHG IN Y FROM ELEMENT TO EXIT POINT
            ''CAN NOW COMPUTE THE DISTANCE TO
            ''THE EXIT POINT TO BYPASS THE OBSTACLE
  DIS.TO.OBS = sqrt.f(F.DEL.X**2 + F.DEL.Y**2)
''+++DEBUG MESSAGE -- SHOW DIST FROM EACH OBSTACLE
''PRINT 1 LINE WITH ID, DIS.TO.OBS THUS
''ELEMENT ** IS ***.** FROM AN OBSTACLE

            ''IF THE RESULT IS BETTER THAN THE CURRENT
            ''BEST, AND A VALID OBSTACLE, THE UPDATE
            ''THE CURRENT BEST
  IF (DIS.TO.OBS < DISTANCE) AND (DIS.TO.OBS >.01)

''+++DEBUG MESSAGE -- SHOW DIST FOR CLOSEST OBS
''PRINT 1 LINE WITH ID, DIS.TO.OBS THUS
''ELEMENT ** IS ***.** METERS FROM AN OBSTACLE

    DISTANCE = DIS.TO.OBS
    OBS.ID = K
  ENDIF
ENDIF

''+++DEBUG MESSAGE -- USED TO MAKE SURE STATUS IS 1
''PRINT 1 LINE WITH ID, ELEMENT.STATUS(V(ID)) THUS
''ELEMENT ** HAS STATUS **
```

```
RETURN
END ''DISTANCE.TO.OBS
```

J. ROUTINE DELTA.DISTANCE

```
ROUTINE DELTA.DISTANCE GIVEN ID,TRAVEL.TIME YIELDING DIS-
TANCE
''THIS ROUTINE DETERMINES THE CHANGE DISTANCE

  DEFINE ID AS AN INTEGER VARIABLE
  DEFINE TRAVEL.TIME,DISTANCE,VELOCITY AS REAL VARIABLES

  VELOCITY = SPEED(V(ID))
  DISTANCE = TRAVEL.TIME * VELOCITY

RETURN
END


ROUTINE DELTA.TIME GIVEN ALLEY,ELEMENT.ID AND DISTANCE
                        YIELDING DURATION

''THIS ROUTINE DETERMINES HOW LONG IT WILL TAKE AN ELEMENT
''TO COVER A GIVEN DISTANCE. THE ROUTINE ACCESSES THE
''ELEMENT RECORD TO DETERMINE ELEMENT SPEED AND RETURNS TO
''THE CALLER THE TIME IT TAKES THAT ELEMENT TO COVER THAT
''DISTANCE.

DEFINE ELEMENT.ID AS AN INTEGER VARIABLE
DEFINE DISTANCE,DURATION AS REAL VARIABLES
IF ALLEY GE 2
  FOR I = 1 TO (ALLEY-1) DO
```

```
    AAA = AAA + N.ELEMENT(ALLEY-I)
  LOOP
  AAA = ELEMENT.ID - AAA
  DURATION = (DISTANCE / SPEED(V(ELEMENT.ID)))*AAA
ELSE
  DURATION = (DISTANCE / SPEED(V(ELEMENT.ID)))*ELEMENT.ID
ALWAYS
  RETURN
END
```

K. ROUTINE FIND.CURRENT.CP

```
ROUTINE FIND.CURRENT.CP GIVEN ID YIELDING CURR.CP

''THIS ROUTINE DETERMINES WHICH CHECKPOINT ON THE
''MOVEMENT.PLAN IS CURRENT FOR A GIVEN ELEMENT

DEFINE ID, AND CURR.CP AS INTEGER VARIABLES
DEFINE J AS AN INTEGER VARIABLE

IF ((ELEMENT.STATUS(V(ID))="ACTIVE")OR(ELEMENT.STATUS(V(ID)) =
                                  "OBSTACLE"))
  FOR J = 1 TO NUM.CP DO          ''CHECK EACH CHECKPOINT IN ORDER,
                       ''LOOKING FOR THE CURRENT ONE FOR
                       ''ELEMENT ID.
    IF MOVEMENT.PLAN(ID,J,5) > 0,
                       ''WHEN FOUND, RECORD THE INDEX
      CURR.CP = J
                       ''AND EXIT THE LOOP
      LEAVE
    ENDIF
  LOOP
ENDIF
```

```
IF ELEMENT.STATUS(V(ID)) = "BYPASS"
 FOR J = 1 TO 4 DO
   IF BYPASS.MAP(ID,J,5) > 0,    ''LOOK FOR ACTIVE STATUS FLAG
                        ''[BYPASS.MAP(*,*,5) > 0]

''+ + + + + + + DEBUG MESSAGE  - RECORDS BYPASS LOCATION
''PRINT 2 LINES WITH ID,J,ELEMENT.X(V(ID)),
''ELEMENT.Y(V(ID)), TIME.V  THUS
''ELEMENT ** IN USING BYPASS CHECKPOINT **
''LOCATION IS X = ***.**  Y = ***.** AT TIME ***.**

     CURR.CP = J
                     ''WHEN FOUND, RECORD THAT POINT AND
     LEAVE              ''EXIT LOOP
   ENDIF
 LOOP
ENDIF

RETURN                      ''RETURN CURRENT CP INDEX
END ''FIND.CURRENT.CP
```

L. ROUTINE OBSTACLE.CONSOLIDATION

ROUTINE OBSTACLE.CONSOLIDATION ''THIS ROUTINE COMBINES TWO OR MORE OBSTACLES INTO A ''SINGLE LARGE OBSTACLE TO FACILITATE BYPASSING

```
DEFINE DISTANCE AS A REAL VARIABLE   ''DISTANCE BET OBSTACLES
DEFINE RAD1 AS A REAL VARIABLE  ''OBSTACLE RADIUS
DEFINE X1, Y1,Z1,Z2 AS REAL VARIABLES
DEFINE RAD2 AS A REAL VARIABLE  ''OBSTACLE RADIUS
DEFINE X2, Y2 AS REAL VARIABLES
```

```
DEFINE CP.RESULT AS AN INTEGER VARIABLE

''+ + + + + + + + + + + + +DEBUG MESSAGE -- RECORD ROUTINE ENTRY
''PRINT 1 LINE THUS
''OBSTACLE.CONSOLIDATION CALLED

'RESTART'
FOR EACH ELEMENT OF OBSTACLE.LIST DO
                ''IF OBSTACLE EXISTS, RECORD NEEDED INFO
 IF ELEMENT.Y > 0
  RAD1 = ELEM.RADIUS
  X1 = ELEMENT.X
  Y1 = ELEMENT.Y
  XX = ELEMENT.NUM
                  ''COMPARE TO ALL REMAINING OBSTACLES
  FOR EACH ELEMENT OF OBSTACLE.LIST WITH ELEMENT.NUM NE XX
DO
    IF ELEMENT.Y > 0
     RAD2 = ELEM.RADIUS
     X2 = ELEMENT.X
     Y2 = ELEMENT.Y
               ''COMPUTE DISTANCE BETWEEN OBSTACLE CENTERS
     DISTANCE = SQRT.F((X1-X2)**2 + (Y1-Y2)**2)
''LIST DISTANCE,XX,ELEMENT.NUM,2*(RAD1+RAD2)
     IF DISTANCE < 2*(RAD1 + RAD2)
      REMOVE ELEMENT FROM OBSTACLE.LIST
      FILE ELEMENT IN RED.TGT.LIST
''+ + + + + + + + + +DEBUG MESSAGE - RECORD WHO IS BEING
''COMBINED PRINT 1 LINE WITH XX,ELEMENT.NUM THUS
''-- COMBINING OBSTACLE ** AND **

      ELEMENT.X(V(XX)) = (X1 + X2)/2.0
      ELEMENT.Y(V(XX)) = (Y1 + Y2)/2.0
      ELEM.RADIUS(V(XX)) = ((DISTANCE + RAD1 + RAD2)/2.0)
     FOR EACH ELEMENT OF RED.TGT.LIST WITH ELEMENT.Y LE
```

```
                        ELEMENT.Y(V(XX)) DO
    BYPASS.MAP(ELEMENT.NUM,2,I) = 0
    CALL BYPASS GIVING ELEMENT.NUM,XX,1.0 YIELDING Z1, Z2
   LOOP

''+ + + + + + + + + + + +DEBUG MESSAGE --RECORD NEW RADIUS
''PRINT 1 LINE WITH OBS.RADIUS(OB(I)) THUS
''-- RESULTING OBSTACLE RADIUS IS ***.**

''UPDATE OBSTACLE.MAP AND FIND NEAREST CP
   CP.RESULT = TRUNC.F(ELEMENT.Y(V(XX))/20) + 1
   OBSTACLE.MAP(XX,CP.RESULT) = 1
            ''TEMP UPDATE PROCEDURE TO CHECK
            ''OBSTACLE MAP - NEEDS TO CHECK
            ''GREATER RANGE.....
   IF (((ELEMENT.Y(V(XX)) - ELEM.RADIUS(V(XX))) <
     MOVEMENT.PLAN(XX,CP.RESULT,2)) AND (CP.RESULT > 1))
    OBSTACLE.MAP(XX,CP.RESULT - 1) = 2
   ENDIF
     ''ELIMINATE 2ND OBSTACLE FROM OBSTACLE LIST
   GO TO 'RESTART'
  ENDIF
 ENDIF
 LOOP
ENDIF
LOOP
RETURN
END
```

M. EVENT OBSTACLE.ENCOUNTER

```
EVENT OBSTACLE.ENCOUNTER GIVEN LINE,ID
''THIS EVENT ALTERS THE STATUS OF THE BREACHER TO SIGNIFY
```

```
''THE INITIATION OF A BYPASS

DEFINE ID AS AN INTEGER VARIABLE

CALL UPDATE.LOCATION
                ''CHANGE ELEMENT STATUS TO BYPASS
ELEMENT.STATUS(V(ID))="BYPASS"

''PRINT 1 LINE WITH ID THUS
''ELEMENT ** IS BYPASSING
BYPASS.MAP(ID,1,5)=1   ''POINT TO 1ST CHECKPOINT IN BYPASS
CALL NEXT.ENCOUNTER GIVING LINE,ID

RETURN
END ''OBSTACLE.ENCOUNTER
```

N. ROUTINE START.SIMO

```
ROUTINE START.SIMO

START SIMULATION

RETURN          END
```

O. EVENT NEW.CP

```
EVENT NEW.CP GIVEN LINE,ID
''THIS EVENT DETERMINES THE BREACHERS NEW/NEXT CHECKPOINT

DEFINE ID AS AN INTEGER VARIABLE
DEFINE CP.ID AS AN INTEGER VARIABLE
```

```
IF ELEMENT.STATUS(V(ID)) = "OBSTACLE"
  RETURN
ALWAYS

CALL UPDATE.LOCATION

CALL FIND.CURRENT.CP GIVING ID YIELDING CP.ID

                    ''IF ELEMENT IN BYPASS MODE
IF ELEMENT.STATUS(V(ID)) EQ "BYPASS"
       ''AND READY TO RE-ENTER MOVEMENT PLAN FROM BYPASS
 IF CP.ID EQ 3
  ELEMENT.X(V(ID)) = BYPASS.MAP(ID,4,1)
  ELEMENT.Y(V(ID)) = BYPASS.MAP(ID,4,2)
  ELEMENT.STATUS(V(ID)) = "ACTIVE"

''   PRINT 1 LINE WITH ID THUS
''   ELEMENT ** IS RE-ENTERING MOVEMENT PATH

   '' ERASE CHECKPOINT STATUS INDICATOR FOR ELEMENT
  FOR I = 1 TO NUM.CP DO
    MOVEMENT.PLAN(ID,I,5) = 0
  LOOP

    ''DETERMINE WHICH CHECKPOINT WAS RE-ENTERED INTO
    ''AND MARK THAT ONE AS THE CURRENT CHECKPOINT
  FOR I = 1 TO NUM.CP-1 DO
    IF ((CP(ID,I) < ELEMENT.Y(V(ID))) AND (CP(ID,I+1) >
                              ELEMENT.Y(V(ID))))
     MOVEMENT.PLAN(ID,I,5) = 1
    ALWAYS
  LOOP
  IF ELEMENT.STATUS(V(ID)) NE "OBSTACLE" AND GRAPHICS = "YES"
   LOCATION.A(V(ID)) = LOCATION.F(ELEMENT.X(V(ID)),
                              ELEMENT.Y(V(ID)))
```

```
    DISPLAY V(ID)
   ALWAYS
   CALL NEXT.ENCOUNTER GIVING LINE,ID
   RETURN
 ALWAYS

 IF CP.ID LT 3
   ELEMENT.X(V(ID)) = BYPASS.MAP(ID,CP.ID+1,1)
   ELEMENT.Y(V(ID)) = BYPASS.MAP(ID,CP.ID+1,2)
 ALWAYS

 "+ + + + + + +DEBUG MESSAGE -- LOCATION ON BYPASS PATH
 " PRINT 1 LINE WITH ID,CP.ID, CP.ID + 1 THUS
 " ELEMENT ** MOVED FROM BYPASS ** TO BYPASS **

 BYPASS.MAP(ID,CP.ID, 5) = 0
 BYPASS.MAP(ID,CP.ID+1, 5) = 1
ALWAYS

IF ELEMENT.STATUS(V(ID)) = "ACTIVE"
                          "ALL DONE MOVING
 IF CP.ID + 1  = NUM.CP
    SPEED(V(ID)) = 0
LIST ID,NUM.CP,ELEMENT.Y(V(ID))
    ELEMENT.STATUS(V(ID)) = "THROUGH"
    RETURN

 ENDIF

                   "UPDATE CHECKPOINT STATUS FLAGS
 MOVEMENT.PLAN(ID,CP.ID,5) = 0
 MOVEMENT.PLAN(ID,CP.ID + 1,5) = 1
```

```
            ''UPDATE ELEMENT POSITION LOCATION
 ELEMENT.X(V(ID)) = MOVEMENT.PLAN(ID,CP.ID + 1,1)
 ELEMENT.Y(V(ID)) = MOVEMENT.PLAN(ID,CP.ID + 1,2)

ENDIF
IF ELEMENT.STATUS(V(ID)) NE "OBSTACLE" AND GRAPHICS = "YES"
 LOCATION.A(V(ID)) = LOCATION.F(ELEMENT.X(V(ID)),
                                ELEMENT.Y(V(ID)))
 DISPLAY V(ID)
ALWAYS


CALL NEXT.ENCOUNTER GIVING LINE,ID

RETURN
END ''NEW.CP
```

P. EVENT MINE.ENCOUNTER

```
EVENT MINE.ENCOUNTER GIVEN LINE,E,M.NUMBER
''THIS EVENT ASSESSES DAMAGE TO BREACHER AND MINE AS A
''RESULT OF A MINE ENCOUNTER BY THE BREACHER

DEFINE
MINE.ID,    ''MINE NUMBER
E           ''BREACHER NUMBER
AS INTEGER VARIABLES

DEFINE
ROLL,  ''MONTE CARLO RESULT
PK     ''FROM USER DEFINED PK TABLES
AS REAL VARIABLES
```

```
DEFINE SEED1 TO MEAN INT.F(RANDOM.F(6)*10)

''TO AVOID 0 SEED
SEED = SEED1
IF SEED = 0
 SEED = 1
ALWAYS

''RETURN IF BREACHER IS DISABLED
IF ELEMENT.STATUS(V(E)) EQ "OBSTACLE"
 RETURN
ALWAYS

CALL UPDATE.LOCATION
IF GRAPHICS EQ "YES"
 LOCATION.A(V(E)) = LOCATION.F(ELEMENT.X(V(E)),ELEMENT.Y(V(E)))
 DISPLAY V(E)
ALWAYS

MINE.ID = 1
IF ELEMENT.TYPE(V(E)) = 1
  IF MINE.ID = 1
    PK = ONE.ONE
  ENDIF
  IF MINE.ID = 2
    PK = ONE.TWO
  ENDIF
  IF MINE.ID = 3
    PK = ONE.THREE
  ENDIF
ENDIF

IF ELEMENT.TYPE(V(E)) = 2
```

```
   IF MINE.ID = 1
     PK = TWO.ONE
   ENDIF
   IF MINE.ID = 2
     PK = TWO.TWO
   ENDIF
   IF MINE.ID = 3
     PK = TWO.THREE
   ENDIF
 ENDIF
 IF ELEMENT.TYPE(V(E)) = 3
   IF MINE.ID = 1
     PK = THREE.ONE
   ENDIF
   IF MINE.ID = 2
     PK = THREE.TWO
   ENDIF
   IF MINE.ID = 3
     PK = THREE.THREE
   ENDIF
 ENDIF
  ROLL = UNIFORM.F(0.0,1.0,SEED)
''   PRINT 1 LINE WITH ROLL AND PK THUS
''   THE ROLL WAS *.***, PK WAS *.***

  IF ROLL < PK
    SKIP 1 LINE
    PRINT 1 LINE WITH E AND M.NUMBER,TIME.V - COMM.TIME USING
    UNIT 2 THUS
     ELEMENT ** DESTROYED BY MINE ***** AT ****.**

     SUBTRACT 1 FROM TOT.ELEMENT
     REMOVE V(E) FROM RED.TGT.LIST
     ELEMENT.STATUS(V(E)) = "OBSTACLE"
     FILE V(E) IN THE OBSTACLE.LIST
```

```
   IF GRAPHICS EQ "YES"
    ERASE V(E)
   ALWAYS
   CALL FIND.CURRENT.CP GIVING E YIELDING CURR.CP
   OBSTACLE.MAP(E,CURR.CP) = 1
   IF (ELEMENT.Y(V(E)) - ELEM.RADIUS(V(E))) <
                            MOVEMENT.PLAN(E,CURR.CP,2)
    IF CURR.CP > 1
     OBSTACLE.MAP(E,CURR.CP - 1) = 2
    ALWAYS
   ALWAYS

   CALL OBSTACLE.CONSOLIDATION
 ELSE
 ''  PRINT 1 LINE WITH E AND M.NUMBER USING UNIT 2 THUS
 ''  ELEMENT ** SURVIVED ENCOUNTER WITH MINE *****

  CALL NEXT.ENCOUNTER GIVING LINE,E
 ENDIF
FOR EACH MINE IN MINE.FIELD WITH MINE.NUMBER EQ M.NUMBER
DO
  IF GRAPHICS EQ "YES"
   DCOLOR.A(ICON.A(MINE)) = 6
   DISPLAY MINE AT (MINE.X,MINE.Y)
   CREATE A DEAD.TANK CALLED DT(E)
   DISPLAY DT(E) WITH "TANK2.ICN" AT (ELEMENT.X(V(E)),
                                ELEMENT.Y(V(E)))
  ALWAYS
  SUBTRACT 1 FROM TOT.ACTIVE
  MINE.STATUS = "HIT BY BREACHER"
 LOOP
 RETURN
 END
```

Q. ROUTINE INITIALIZE

```
ROUTINE INITIALIZE
''THIS ROUTINE INPUTS USER DEFINED PARAMETERS AND
''INITIALIZES SIMULATION

  DEFINE ELEMENT.SPEED AS A REAL VARIABLES
  DEFINE SEED1 TO MEAN INT.F(RANDOM.F(1)*10)

''TO AVOID 0 SEED
  SEED = SEED1
  IF SEED = 0
   SEED = 1
  ALWAYS


  ''LOAD INPUT FORM AND SET DEFAULT DATA
  SHOW INPUT.FORM WITH "KHAFJI.FRM"
  LET DTVAL.A(DFIELD.F("GZ",INPUT.FORM)) = "NX"
  LET DDVAL.A(DFIELD.F("GZE",INPUT.FORM)) = 56
  LET DDVAL.A(DFIELD.F("GZN",INPUT.FORM)) = 78
  LET DDVAL.A(DFIELD.F("N.BELT",INPUT.FORM)) = 1
  LET DDVAL.A(DFIELD.F("M.WIDTH",INPUT.FORM)) = 200
  LET DDVAL.A(DFIELD.F("RADIUS", INPUT.FORM)) = .2
  LET DDVAL.A(DFIELD.F("DIST", INPUT.FORM)) = 100.0
  LET DDVAL.A(DFIELD.F("DEV",INPUT.FORM)) = 5
  LET DDVAL.A(DFIELD.F("N.LANE",INPUT.FORM)) = 2
  LET DDVAL.A(DFIELD.F("TOT.ELEMENT",INPUT.FORM)) = 7
  LET DTVAL.A(DFIELD.F("GO",INPUT.FORM)) = "YES"
  LET DDVAL.A(DFIELD.F("N.WEAPON",INPUT.FORM)) = 2

  CALL DEVINIT.R("VT,GRAPHIC") YIELDING DEVPTR
  OPEN 7 FOR INPUT, DEVICE = DEVPTR
  OPEN 8 FOR OUTPUT, DEVICE = DEVPTR
```

```
USE 8 FOR GRAPHIC OUTPUT

LET FIELD.ID = ACCEPT.F(INPUT.FORM,0)

LET TOT.ELEMENT = DDVAL.A(DFIELD.F("TOT.ELEMENT",
                              INPUT.FORM))
LET N.WEAPON = DDVAL.A(DFIELD.F("N.WEAPON",INPUT.FORM))
LET N.LANE = DDVAL.A(DFIELD.F("N.LANE",INPUT.FORM))
LET GR.ZONE = DTVAL.A(DFIELD.F("GZ",INPUT.FORM))
LET GR.BASE.E = DDVAL.A(DFIELD.F("GZE",INPUT.FORM))
LET GR.BASE.N = DDVAL.A(DFIELD.F("GZN",INPUT.FORM))
LET N.MINE.BELT = DDVAL.A(DFIELD.F("N.BELT",INPUT.FORM))
LET MF.WIDTH = DDVAL.A(DFIELD.F("M.WIDTH",INPUT.FORM))
LET MINE.DEV = DDVAL.A(DFIELD.F("DEV",INPUT.FORM))
LET TEMP.RADIUS = DDVAL.A(DFIELD.F("RADIUS", INPUT.FORM))
LET MAX.DISTANCE  = DDVAL.A(DFIELD.F("DIST", INPUT.FORM))
LET GO =  DTVAL.A(DFIELD.F("GO",INPUT.FORM))


SHOW TANK.FORM WITH "TANK.FRM"
                              ''METERS
LET DDVAL.A(DFIELD.F("TANK.WIDTH", TANK.FORM)) = 3.4798
                    ''LENGTH WITH GUN FORWARD
LET DDVAL.A(DFIELD.F("TANK.LENGTH", TANK.FORM)) = 9.828276
LET DDVAL.A(DFIELD.F("TRACK.WIDTH", TANK.FORM)) = .635
LET DDVAL.A(DFIELD.F("BREACHER.TYPE", TANK.FORM)) = 1
LET DDVAL.A(DFIELD.F("KILL.RATE", TANK.FORM)) = .03
LET DDVAL.A(DFIELD.F("SPEED", TANK.FORM)) = 85

LET FIELD.ID = ACCEPT.F(TANK.FORM,0)

LET E.WIDTH     = DDVAL.A(DFIELD.F("TANK.WIDTH", TANK.FORM))
LET E.LENGTH    = DDVAL.A(DFIELD.F("TANK.LENGTH", TANK.FORM))
LET E.TRACK.WIDTH = DDVAL.A(DFIELD.F("TRACK.WIDTH",
                                TANK.FORM))
```

```
LET BREACHER.TYPE = DDVAL.A(DFIELD.F("BREACHER.TYPE",
                        TANK.FORM))
LET KILL.RATE = DDVAL.A(DFIELD.F("KILL.RATE", TANK.FORM))
LET ELEMENT.SPEED = DDVAL.A(DFIELD.F("SPEED", TANK.FORM))


SHOW PK.FORM WITH "PKFORM.FRM"
LET DDVAL.A(DFIELD.F("ONE-ONE", PK.FORM)) = 1
LET DDVAL.A(DFIELD.F("ONE-TWO", PK.FORM)) = 1
LET DDVAL.A(DFIELD.F("ONE-THREE", PK.FORM)) = 1
LET DDVAL.A(DFIELD.F("TWO-ONE", PK.FORM)) = 1
LET DDVAL.A(DFIELD.F("TWO-TWO", PK.FORM)) = 1
LET DDVAL.A(DFIELD.F("TWO-THREE", PK.FORM)) = 1
LET DDVAL.A(DFIELD.F("THREE-ONE", PK.FORM)) = 1
LET DDVAL.A(DFIELD.F("THREE-TWO", PK.FORM)) = 1
LET DDVAL.A(DFIELD.F("THREE-THREE", PK.FORM)) = 1

LET FIELD.ID = ACCEPT.F(PK.FORM,0)

LET ONE.ONE     = DDVAL.A(DFIELD.F("ONE-ONE", PK.FORM))
LET ONE.TWO     = DDVAL.A(DFIELD.F("ONE-TWO", PK.FORM))
LET ONE.THREE   = DDVAL.A(DFIELD.F("ONE-THREE", PK.FORM))
LET TWO.ONE     = DDVAL.A(DFIELD.F("TWO-ONE", PK.FORM))
LET TWO.TWO     = DDVAL.A(DFIELD.F("TWO-TWO", PK.FORM))
LET TWO.THREE   = DDVAL.A(DFIELD.F("TWO-THREE", PK.FORM))
LET THREE.ONE   = DDVAL.A(DFIELD.F("THREE-ONE", PK.FORM))
LET THREE.TWO   = DDVAL.A(DFIELD.F("THREE-TWO", PK.FORM))
LET THREE.THREE = DDVAL.A(DFIELD.F("THREE-THREE", PK.FORM))

CREATE EACH MINE.BELT
FOR EACH MINE.BELT DO
  SHOW INPUT2.FORM WITH "KHAFJI2.FRM"
  LET DDVAL.A(DFIELD.F("MINE.BELT",INPUT2.FORM)) = MINE.BELT
  LET DDVAL.A(DFIELD.F("N.MINES(MINE.BELT)",INPUT2.FORM)) = 50
  LET DDVAL.A(DFIELD.F("DEPTH(MINE.BELT)",INPUT2.FORM)) = 50
```

```
  LET DDVAL.A(DFIELD.F("MINE.TYPE",INPUT2.FORM)) = 1

  LET FIELD.ID = ACCEPT.F(INPUT2.FORM,0)

  LET MINE.B.TYPE(MINE.BELT) =
       DDVAL.A(DFIELD.F("MINE.TYPE",INPUT2.FORM))
  LET DEPTH(MINE.BELT) =
       DDVAL.A(DFIELD.F("DEPTH(MINE.BELT)",INPUT2.FORM))
  LET N.MINES(MINE.BELT) =
       DDVAL.A(DFIELD.F("N.MINES(MINE.BELT)",INPUT2.FORM))
LOOP

CREATE EACH WEAPON
FOR EACH WEAPON DO
  SHOW INPUT4.FORM WITH "KHAFJI4.FRM"
  LET DDVAL.A(DFIELD.F("WEAPON",INPUT4.FORM)) = WEAPON
  LET DTVAL.A(DFIELD.F("WEAPON.TYPE",INPUT4.FORM))=
                              "HOWITZER"
  LET DTVAL.A(DFIELD.F("WEAPON.UNIT",INPUT4.FORM)) = "BLUE"
  LET DDVAL.A(DFIELD.F("LETHAL.RADIUS",INPUT4.FORM)) = 10
  LET DDVAL.A(DFIELD.F("ERROR.X",INPUT4.FORM)) = 15
  LET DDVAL.A(DFIELD.F("ERROR.Y",INPUT4.FORM)) = 15
  LET DDVAL.A(DFIELD.F("N.VOLLEYS",INPUT4.FORM)) = 4

  LET FIELD.ID = ACCEPT.F(INPUT4.FORM,0)

  LET WEAPON.TYPE=
       DTVAL.A(DFIELD.F("WEAPON.TYPE",INPUT4.FORM))
  LET WEAPON.UNIT=
       DTVAL.A(DFIELD.F("WEAPON.UNIT",INPUT4.FORM))
  LET LETHAL.RADIUS=DDVAL.A(DFIELD.F("LETHAL.RADIUS",
                              INPUT4.FORM))
  LET ERROR.X = DDVAL.A(DFIELD.F("ERROR.X",INPUT4.FORM))
  LET ERROR.Y = DDVAL.A(DFIELD.F("ERROR.Y",INPUT4.FORM))
  LET N.VOLLEYS = DDVAL.A(DFIELD.F("N.VOLLEYS",INPUT4.FORM))
```

```
 LOOP
   OPEN UNIT 1 FOR INPUT, NAME IS "WEAPONS.DAT"
   OPEN UNIT 2 FOR OUTPUT, NAME IS "OUTFILE.DAT"
   USE 2 FOR OUTPUT

   GR.BASE.E = GR.BASE.E*1000
   GR.BASE.N = GR.BASE.N*1000

   READ COMM.TIME,FDC.TIME,INIT.GUN.TIME,
      GUN.TIME.PER.ROUND,TOF USING UNIT 1

RESERVE V(*) AS TOT.ELEMENT
CREATE EACH LANE
FOR EACH LANE DO
 SHOW INPUT3.FORM WITH "KHAFJI3.FRM"
 LET DDVAL.A(DFIELD.F("N.ELEMENT",INPUT3.FORM)) = 4
 LET DDVAL.A(DFIELD.F("LANE",INPUT3.FORM)) = LANE
 LET DDVAL.A(DFIELD.F("START.X",INPUT3.FORM)) = 561

 LET FIELD.ID = ACCEPT.F(INPUT3.FORM,0)

 LET N.ELEMENT = DDVAL.A(DFIELD.F("N.ELEMENT",INPUT3.FORM))
 LET START.X = 100*DDVAL.A(DFIELD.F("START.X",INPUT3.FORM)) -
                                   GR.BASE.E
LOOP

NUM.CP = 2*INT.F((TOT.ELEMENT/N.LANE)+(MAX.DISTANCE / 20.0) +
2.0)

 SHOW INPUT5.FORM WITH "KHAFJI5.FRM"
 LET DTVAL.A(DFIELD.F("GRAPHICS",INPUT5.FORM)) = "YES"
 LET DDVAL.A(DFIELD.F("N.RUN",INPUT5.FORM)) = 5

 LET FIELD.ID = ACCEPT.F(INPUT5.FORM,0)
```

```
LET GRAPHICS = DTVAL.A(DFIELD.F("GRAPHICS",INPUT5.FORM))
LET N.RUN = DDVAL.A(DFIELD.F("N.RUN",INPUT5.FORM))

    FOR I = 1 TO 10 DO
      LET S(I) = SEED.V(I)
    LOOP

RESERVE MOVEMENT.PLAN(*,*,*) AS TOT.ELEMENT BY NUM.CP BY 5
RESERVE OBSTACLE.MAP(*,*) AS TOT.ELEMENT BY MAX.F(NUM.CP,5)
RESERVE CP(*,*) AS TOT.ELEMENT BY NUM.CP
RESERVE BYPASS.MAP(*,*,*) AS TOT.ELEMENT BY 4 BY 5
RESERVE DT(*) AS TOT.ELEMENT

FOR Z = 1 TO N.LANE DO
 IF Z = 1
  FOR I = 1 to N.ELEMENT(Z) DO
    CREATE A ELEMENT CALLED V(I)
    ELEMENT.STATUS(V(I)) = "ACTIVE"
    FILE V(I) IN THE RED.TGT.LIST
    ELEMENT.NUM(V(I)) = I
    CALL MAKE.ROUTE
              ''INITIALIZE ELEMENT START POINT LOCATION
              ''BASED ON MOVEMENT PLAN
    ELEMENT.X(V(I)) = MOVEMENT.PLAN(I,1,1)
    ELEMENT.Y(V(I)) = MOVEMENT.PLAN(I,1,2)
    IF GO = "YES"
     SPEED(V(I)) = ELEMENT.SPEED*(1.0 - (N.VOLLEYS*0.05))
    ELSE
     SPEED(V(I)) = ELEMENT.SPEED      ''METERS PER MINUTE
    ALWAYS
    SHOW V(I) WITH "TANK"
    WIDTH(V(I)) = E.WIDTH
    ELEM.RADIUS(V(I)) = SQRT.F((WIDTH(V(I))*E.LENGTH)/PI.C)
    TRACK.WIDTH(V(I)) = E.TRACK.WIDTH
    ELEMENT.TYPE(V(I)) = BREACHER.TYPE
```

```
      DEATH.TIME(V(I)) = NORMAL.F(1/KILL.RATE,5.0,SEED)
      CALL NEXT.ENCOUNTER GIVING Z,I
    LOOP
  ELSE
    AAA = 0
    FOR I = 1 TO Z - 1 DO
      AAA =  AAA + N.ELEMENT(Z-I)
    LOOP
    FOR I = (AAA+1) to (N.ELEMENT(Z) + AAA) DO
      CREATE A ELEMENT CALLED V(I)
      ELEMENT.STATUS(V(I)) = "ACTIVE"
      FILE V(I) IN THE RED.TGT.LIST
      ELEMENT.NUM(V(I)) = I
      call MAKE.ROUTE
                  ''INITIALIZE ELEMENT START POINT LOCATION
                  ''BASED ON MOVEMENT PLAN
      ELEMENT.X(V(I)) = MOVEMENT.PLAN(I,1,1)
      ELEMENT.Y(V(I)) = MOVEMENT.PLAN(I,1,2)
      IF GO = "YES"
        SPEED(V(I)) = ELEMENT.SPEED*(1.0 - (N.VOLLEYS*0.05))
      ELSE
        SPEED(V(I)) = ELEMENT.SPEED       ''METERS PER MINUTE
      ALWAYS
      SHOW V(I) WITH "TANK"
      WIDTH(V(I)) = E.WIDTH
      ELEM.RADIUS(V(I)) = SQRT.F((WIDTH(V(I))*E.LENGTH)/PI.C)
      TRACK.WIDTH(V(I)) = E.TRACK.WIDTH
      ELEMENT.TYPE(V(I)) = BREACHER.TYPE
      DEATH.TIME(V(I)) = NORMAL.F(1/KILL.RATE,5.0,SEED)
      CALL NEXT.ENCOUNTER GIVING Z,I
    LOOP
  ALWAYS
LOOP

FOR I = 1 TO TOT.ELEMENT DO
```

```
    LOCATION.A(V(I)) = LOCATION.F(ELEMENT.X(V(I)),ELEMENT.Y(V(I)))
LOOP


RETURN  END
```

R. ROUTINE UPDATE.LOCATION

```
ROUTINE UPDATE.LOCATION

''THIS ROUTINE CAUSES THE IDENTIFIED ELE TO BE MOVED ALONG ITS
''PREDISIGNATED MOVEMENT PATH A SPECIFIED DISTANCE. BY
''FIRST DETERMINING WHAT PATH TO USE,CALC COORDINATES
''OF THE MOVE RESULT, THEN UPDATING ELE LOCATION FIELDS

 DEFINE DISTANCE AS A REAL VARIABLE          ''DISTANCE TO NEXT CP
 DEFINE DISTANCE.TO.TRAVEL AS A REAL VARIABLE ''TRVL DIST
 DEFINE TIME.PASSAGE AS REAL VARIABLES
 DEFINE X1 AS A REAL VARIABLE                ''DELTA X TO NEXT CP
 DEFINE Y1 AS A REAL VARIABLE                ''DELTA Y TO NEXT CP
 DEFINE X2 AS A REAL VARIABLE                ''MOVEMENT IN X
 DEFINE Y2 AS A REAL VARIABLE                ''MOVEMENT IN Y
 DEFINE MOV.FRACTION AS A REAL VARIABLE ''ELE MOVEMENT AS
                                ''A FRACTION OF DISTANCE
                                ''TO NEXT CHECKPOINT
''PRINT 1 LINE WITH TIME.V, OLD.TIME USING UNIT 2 THUS
''LOCATION UPDATE IS OCCURING AT ***.** LAST UPDATE WAS AT ***.**

IF TOT.ELEMENT LE 0
  RETURN
ALWAYS

LET TIME.PASSAGE = TIME.V - OLD.TIME
```

```
LET OLD.TIME = TIME.V

  FOR EACH ELEMENT OF RED.TGT.LIST WITH ELEMENT.STATUS EQ
              "ACTIVE" OR ELEMENT.STATUS EQ "BYPASS" DO
    IF TIME.V GE DEATH.TIME AND ELEMENT.Y GT 20
      SUBTRACT 1 FROM TOT.ELEMENT
      REMOVE ELEMENT FROM RED.TGT.LIST
      ELEMENT.STATUS = "OBSTACLE"
      FILE ELEMENT IN THE OBSTACLE.LIST
      IF GRAPHICS EQ "YES"
        ERASE ELEMENT
        CREATE A DEAD.TANK CALLED DT(ELEMENT.NUM)
        DISPLAY DT(ELEMENT.NUM) WITH "TANK2.ICN" AT
                          (ELEMENT.X,ELEMENT.Y)
      ALWAYS
      CALL FIND.CURRENT.CP GIVING ELEMENT.NUM YIELDING
                              CURR.CP
      OBSTACLE.MAP(ELEMENT.NUM,CURR.CP) = 1

      IF (ELEMENT.Y - ELEM.RADIUS) <
                    MOVEMENT.PLAN(ELEMENT.NUM,CURR.CP,2)
        IF CURR.CP > 1
          OBSTACLE.MAP(ELEMENT.NUM,CURR.CP - 1) = 2
        ALWAYS
      ALWAYS
      SKIP 2 LINES
      PRINT 1 LINE WITH ELEMENT.NUM, TIME.V - COMM.TIME THUS
      ELEMENT *** KILLED BY DIRECT FIRE AT ****.**

      CALL OBSTACLE.CONSOLIDATION
      RETURN
    ALWAYS
```

```
    CALL DELTA.DISTANCE GIVING ELEMENT.NUM AND TIME.PASSAGE
                        YIELDING DISTANCE.TO.TRAVEL

              ''DETERMINE DISTANCE TO NEXT CHECKPOINT
              ''AND THE DELTA X, DELTA Y
    CALL DISTANCE.TO.CP GIVING ELEMENT.NUM YIELDING
                             DISTANCE, X1, Y1


            ''COMPUTE THE FRACTION OF THE DISTANCE
            ''TO THE NEXT CHECKPOINT THAT THE REQUIRED
            ''MOVE WILL COVER
    MOV.FRACTION = DISTANCE.TO.TRAVEL / DISTANCE

              ''TRANSLATE THAT FRACTION INTO X AND Y
              ''MOVEMENT
    X2 = X1 * MOV.FRACTION
    Y2 = Y1 * MOV.FRACTION
              ''AND ADD THAT MOVEMENT TO THE CURRENT
              ''ELEMENT POSITION
    ELEMENT.X = ELEMENT.X + X2
    ELEMENT.Y = ELEMENT.Y + Y2
    IF ELEMENT.Y GT 2*MAX.DISTANCE
     ELEMENT.STATUS = "THROUGH"
      SUBTRACT 1 FROM TOT.ELEMENT ''LIST
ELEMENT.NUM,ELEMENT.STATUS,TOT.ELEMENT,ELEMENT.Y
    ALWAYS

  LOOP
RETURN
END
```

S. PROCESS RED.ARTY.ATK

```
PROCESS RED.ARTY.ATK GIVEN AIMPOINT.X AND AIMPOINT.Y
''THIS PROCESS FIRES RED ARTILLERY AGAINST BREACHING FORCE

 DEFINE IMPACT.X,IMPACT.Y,AIMPOINT.X,AIMPOINT.Y,DISTANCE,
 PVAR, ''POOLED VARIANCE OF WEAPON ERRORS (COOKIE CUTTER)
 XX
  AS REAL VARIABLES
  DEFINE SEED1 TO MEAN INT.F(RANDOM.F(7)*10)

''TO AVOID 0 SEED
  SEED = SEED1
  IF SEED = 0
   SEED = 1
  ALWAYS

''TO COMPUTE POOLED VARIANCE
  FOR EACH WEAPON DO
    IF WEAPON.UNIT = "RED"
     PVAR = SQRT.F(ERROR.X**2+ERROR.Y**2)
     ADD (LETHAL.RADIUS**2)/(2*PVAR) TO XX
    ALWAYS
  LOOP

''PROBABILITY OF KILL (COOKIE CUTTER)
  P.KILL = 1-EXP.F(-XX)


  FOR EACH WEAPON DO
    IF WEAPON.UNIT = "RED"
      FOR I = 1 TO N.VOLLEYS DO
''NORMAL SAMPLING TO DETERMINE IMPACT LOCATION
        IMPACT.X = NORMAL.F(AIMPOINT.X,ERROR.X,SEED)
        IMPACT.Y = NORMAL.F(AIMPOINT.Y,ERROR.Y,SEED)
        ADD 1 TO N.RED.ROUND
```

```
    IF GRAPHICS EQ "YES"
      CREATE AN XPLODE CALLED XR(N.RED.ROUND)
    DISPLAY XR(N.RED.ROUND) WITH "XPLODER.ICN" AT
                    (IMPACT.X,IMPACT.Y)
    ALWAYS

''TO DETERMINE TARGET,BREACHERS AND MINES, PROXIMITY TO IM-
PACT
    FOR EACH MINE OF MINE.FIELD WITH ((IMPACT.X-MINE.X)
      + (IMPACT.Y-MINE.Y)) LE (1.42*LETHAL.RADIUS) AND
                MINE.STATUS EQ "ACTIVE" DO
      DISTANCE=SQRT.F((IMPACT.X-MINE.X)**2 +
                    (IMPACT.Y-MINE.Y)**2)

''TO DETERMINE DAMAGE ASSESSMENT AGAINST MINE
      IF DISTANCE LE LETHAL.RADIUS
    PRINT 1 LINE WITH MINE.NUMBER,WEAPON.UNIT,
    WEAPON.TYPE,TIME.V-COMM.TIME THUS
MINE # **** DESTROYED BY **** ******** AT TIME ****.**, AT GRID:

       PRINT 1 LINE WITH  GR.ZONE,MINE.X+GR.BASE.E,
       MINE.Y+GR.BASE.N USING UNIT 2 THUS
    ** *****.*   *****.*

       SUBTRACT 1 FROM TOT.ACTIVE
       MINE.STATUS =  "DESTROYED BY RED ARTY"
       IF GRAPHICS = "YES"
        DCOLOR.A(ICON.A(MINE)) = 6
        DISPLAY MINE
       ALWAYS
      ALWAYS
     LOOP

''TO DETERMINE DAMAGE ASSESSMENT AGAINST BREACHER
    IF RANDOM.F(1) LE P.KILL
```

```
        FOR EACH ELEMENT OF RED.TGT.LIST WITH
        ((IMPACT.X-ELEMENT.X) + (IMPACT.Y-ELEMENT.Y)) LE
                        (1.42*LETHAL.RADIUS) DO
          DISTANCE = SQRT.F((IMPACT.X-ELEMENT.X)**2 +
                        (IMPACT.Y-ELEMENT.Y)**2)
          IF DISTANCE LE LETHAL.RADIUS
           PRINT 1 LINE WITH ELEMENT.NUM, WEAPON.UNIT,
           WEAPON.TYPE THUS
ELEMENT # ** DESTROYED BY **** ******** AT GRID:

           PRINT 1 LINE WITH GR.ZONE,ELEMENT.X+GR.BASE.E,
ELEMENT.Y+GR.BASE.N USING UNIT 2 THUS
      ** *****.*   *****.*

           SUBTRACT 1 FROM TOT.ELEMENT
           REMOVE ELEMENT FROM RED.TGT.LIST
           ELEMENT.STATUS = "OBSTACLE"
           IF ELEMENT.Y GT 0  ''IN MINE FIELD
            FILE ELEMENT IN THE OBSTACLE.LIST
           ALWAYS
           IF GRAPHICS EQ "YES"
            CREATE A DEAD.TANK CALLED DT(ELEMENT.NUM)
DISPLAY    DT(ELEMENT.NUM)    WITH    "TANK2.ICN"    AT
(ELEMENT.X,ELEMENT.Y)
            ERASE ELEMENT
           ALWAYS
          ALWAYS
        LOOP
      ALWAYS
    LOOP
  ALWAYS
 LOOP

 PRINT 2 LINE WITH INITIAL - TOT.ACTIVE, TOT.ACTIVE THUS
MINES DESTROYED ****
```

```
MINES REMAINING ****


RETURN          END
```

T. ROUTINE BYPASS

```
ROUTINE BYPASS GIVEN ID, OBS.ID, C.X YIELDING X1, Y1


DEFINE OBS.ID AND ID AS INTEGER VARIABLES
DEFINE C.X AS A REAL VARIABLE
DEFINE OBSTACLE.RADIUS AS A REAL VARIABLE
DEFINE ELEMENT.RADIUS AS A REAL VARIABLE
DEFINE COMBINED.RADIUS AS A REAL VARIABLE
DEFINE X.OB AS A REAL VARIABLE
DEFINE Y.OB AS A REAL VARIABLE
DEFINE BYPASS.X AS A REAL VARIABLE
DEFINE HYPOT AS A REAL VARIABLE
DEFINE BYPASS.SEG AS A REAL VARIABLE
DEFINE PIVOT.1.Y AS A REAL VARIABLE
DEFINE PIVOT.2.Y AS A REAL VARIABLE
DEFINE CURR.CP AS AN INTEGER VARIABLE
DEFINE PATH.1.SLOPE AS A REAL VARIABLE
DEFINE PATH.1.INTERCEPT AS A REAL VARIABLE
DEFINE PATH.2.SLOPE AS A REAL VARIABLE
DEFINE PATH.2.INTERCEPT AS A REAL VARIABLE
DEFINE LINE.1.SLOPE AS A REAL VARIABLE
DEFINE LINE.1.INTERCEPT AS A REAL VARIABLE
DEFINE LINE.2.SLOPE AS A REAL VARIABLE
DEFINE LINE.2.INTERCEPT AS A REAL VARIABLE
DEFINE X1, Y1 AS REAL VARIABLES ''COORDINATES OF EXIT POINT
DEFINE X2, Y2 AS REAL VARIABLES
```

```
DEFINE NEXT.CP.Y AS A REAL VARIABLE
DEFINE FINAL.Y AS A REAL VARIABLE
DEFINE J AS AN INTEGER VARIABLE


IF BYPASS.MAP(ID,2,1) = BYPASS.X NE 0     ''PIVOT.1 X COORDINATE
  RETURN
ALWAYS

OBSTACLE.RADIUS = ELEM.RADIUS(V(OBS.ID))
ELEMENT.RADIUS = ELEM.RADIUS(V(ID))
COMBINED.RADIUS = OBSTACLE.RADIUS + ELEMENT.RADIUS
X.OB = ELEMENT.X(V(OBS.ID))
Y.OB = ELEMENT.Y(V(OBS.ID))
FINAL.Y = CP(ID,NUM.CP)

CALL FIND.CURRENT.CP GIVING ID YIELDING CURR.CP

IF X.OB < C.X                      ''PASS ON THE RIGHT

'' PRINT 1 LINE WITH X.OB, C.X, ID THUS
'' OBSTACLE CENTER IS AT X = ***.**, INT X = ***.** FOR ELEMENT **

 BYPASS.X = X.OB + COMBINED.RADIUS
 LINE.1.SLOPE = 1.0
 LINE.2.SLOPE = -1.0
ENDIF

IF X.OB > = C.X                    ''PASS ON THE LEFT

''  PRINT 1 LINE WITH X.OB, C.X, ID THUS
''  OBSTACLE X = ***.**, INT X = ***.**, ID = **  PASSING LEFT

 BYPASS.X = X.OB - COMBINED.RADIUS
 LINE.1.SLOPE = -1.0
```

```
 LINE.2.SLOPE = 1.0
ENDIF

''  PRINT 1 LINE WITH ID THUS
''  BUILDING A BYPASS MAP FOR ELEMENT **

 HYPOT = SQRT.F(2 * (COMBINED.RADIUS**2))
 BYPASS.SEG = HYPOT - COMBINED.RADIUS
 PIVOT.1.Y = Y.OB - BYPASS.SEG

 BYPASS.MAP(ID,2,1) = BYPASS.X   ''PIVOT.1 X COORDINATE
 BYPASS.MAP(ID,2,2) = PIVOT.1.Y  ''PIVOT.1 Y COORDINATE

 PIVOT.2.Y = Y.OB + BYPASS.SEG

 BYPASS.MAP(ID,3,1) = BYPASS.X   ''PIVOT.2 X COORDINATE
 BYPASS.MAP(ID,3,2) = PIVOT.2.Y  ''PIVOT.2 Y COORDINATE
 BYPASS.MAP(ID,2,3) = 1000000.0  ''BIG NUMBER TO APPROX INFINITY

                         ''INCOMING PATH EQUATION
 PATH.1.SLOPE = MOVEMENT.PLAN(ID,CURR.CP, 3)
 PATH.1.INTERCEPT = MOVEMENT.PLAN(ID,CURR.CP, 4)

 LINE.1.INTERCEPT = PIVOT.1.Y - (LINE.1.SLOPE * BYPASS.X)

                        ''INTERSECTION POINT OF PATH EQN
                         ''AND FIRST LEG OF BYPASS
 X1 = (LINE.1.INTERCEPT - PATH.1.INTERCEPT) /
                          (PATH.1.SLOPE - LINE.1.SLOPE)
 Y1 = PATH.1.SLOPE * X1 + PATH.1.INTERCEPT

 BYPASS.MAP(ID,1,1) = X1         ''EXIT.X COORDINATE
 BYPASS.MAP(ID,1,2) = Y1         ''EXIT.Y COORDINATE
 BYPASS.MAP(ID,1,3) = LINE.1.SLOPE       ''SLOPE FOR LEG 1
 BYPASS.MAP(ID,1,4) = LINE.1.INTTERCEPT  ''Y INTERCEPT FOR LEG 1
```

```
J = 1                        ''INTERVAL INCREMENTER

CALL FIND.CURRENT.CP GIVING ID YIELDING CURR.CP

'RECHECK'

NEXT.CP.Y = MOVEMENT.PLAN(ID,CURR.CP + J, 2)

LINE.2.INTERCEPT = PIVOT.2.Y - (LINE.2.SLOPE * BYPASS.X)

IF NEXT.CP.Y = FINAL.Y                      ''LAST CHECKPOINT
  Y2 = FINAL.Y
  X2 = (FINAL.Y - LINE.2.INTERCEPT) / LINE.2.SLOPE
ENDIF

IF NEXT.CP.Y NE FINAL.Y
  PATH.2.SLOPE = MOVEMENT.PLAN(ID,CURR.CP,3) ''CURR PATH SLOPE
  PATH.2.INTERCEPT  =  MOVEMENT.PLAN(ID,CURR.CP,4)''CURR  PATH
INT

              '' ASSUME STILL IN SAME CP INTERVAL
              '' AND COMPUTE REENTRY POINT
  X2 = (LINE.2.INTERCEPT - PATH.2.INTERCEPT) /
                     (PATH.2.SLOPE - LINE.2.SLOPE)
  Y2 = PATH.2.SLOPE * X2 + PATH.2.INTERCEPT
ENDIF

BYPASS.MAP(ID,4,1) = X2            ''RE-ENTRY X
BYPASS.MAP(ID,4,2) = Y2            ''RE-ENTRY Y
BYPASS.MAP(ID,3,3) = LINE.2.SLOPE
BYPASS.MAP(ID,3,4) = LINE.2.INTERCEPT

            ''IF ENTERENCE INTERCEPT NOT IN THIS INTERVAL
```

```
            ''CHECK THE NEXT INTERVAL
 IF Y2 > NEXT.CP.Y
  J = J + 1             ''INTERVAL INCREMENTER
  GO TO RECHECK         ''GOTO
 ENDIF

RETURN
END ''BYPASS



ROUTINE RESTART

DEFINE SEED1 TO MEAN INT.F(RANDOM.F(8)*10)

SEED = SEED1
IF SEED = 0
 SEED = 1
ALWAYS

FOR Z = 1 TO N.LANE DO
 IF Z = 1
  FOR I = 1 to N.ELEMENT(Z) DO
    FOR J = 1 TO 4 DO
      FOR K = 1 TO 5 DO
        BYPASS.MAP(I,J,K) = 0
      LOOP
    LOOP
    ELEMENT.STATUS(V(I)) = "ACTIVE"
    CALL MAKE.ROUTE
    ELEMENT.X(V(I)) = MOVEMENT.PLAN(I,1,1)
    ELEMENT.Y(V(I)) = MOVEMENT.PLAN(I,1,2)
    ELEM.RADIUS(V(I)) = SQRT.F((WIDTH(V(I))*E.LENGTH)/PI.C)
    DEATH.TIME(V(I)) = NORMAL.F(1/KILL.RATE,5.0,SEED) + TIME.V
    MOVEMENT.PLAN(I,1,5) = 1
    CALL NEXT.ENCOUNTER GIVING Z,I
```

```
    LOOP
  ELSE
    AAA = 0
    FOR I = 1 TO (Z - 1) DO
      AAA = AAA + N.ELEMENT(Z-I)
    LOOP
    FOR I = (AAA + 1) to (N.ELEMENT(Z) + AAA) DO
      ELEMENT.STATUS(V(I)) = "ACTIVE"
      FOR J = 1 TO 4 DO
        FOR K = 1 TO 5 DO
          BYPASS.MAP(I,J,K) = 0
        LOOP
      LOOP
      CALL MAKE.ROUTE
      ELEMENT.X(V(I)) = MOVEMENT.PLAN(I,1,1)
      ELEMENT.Y(V(I)) = MOVEMENT.PLAN(I,1,2)
      ELEM.RADIUS(V(I)) = SQRT.F((WIDTH(V(I))*E.LENGTH)/PI.C)
      DEATH.TIME(V(I)) = NORMAL.F(1/KILL.RATE,5.0,SEED) + TIME.V
      MOVEMENT.PLAN(I,1,5) = 1
      CALL NEXT.ENCOUNTER GIVING Z,I
    LOOP
  ALWAYS
LOOP

FOR I = 1 TO TOT.ELEMENT DO
  LOCATION.A(V(I)) = LOCATION.F(ELEMENT.X(V(I)),ELEMENT.Y(V(I)))
LOOP

RETURN  END
```

U. ROUTINE SET.DISPLAY

```
ROUTINE SET.DISPLAY
''THIS ROUTINE SETS GRAPHICS WINDOW

  VXFORM.V = 1
  CALL SETWORLD.R(0,MF.WIDTH,-MAX.DISTANCE/2,MAX.DISTANCE)
  TIMESCALE.V = 600

RETURN END
```

LIST OF REFERENCES

1. Heckman, W., *Rommel's War in Africa*, Double Day & Company, Inc., Garden City, N.Y., 1981.

2. Barnett, C., *The Desert Generals*, Indiana University Press, Bloomington, IN., 1982.

3. Stolfi, R. H., *Mine and Countermine Warfare in Recent History, 1914-1970*, U.S. Army Ballistic Research Laboratories, Aberdeen Proving Ground, Md., April 72.

4. *U. S. News and World Report*, U. S. News & World Report, Inc., Washington, D.C., 1991.

5. *Janes's Military Vehicles and Ground Support Equipment*, Janes's Publishing Inc., New York, N.Y., 1987.

6. Anderson, A., *Methodologies for the High Resolution Modeling of Mine Dynamics*, Naval Postgraduate School, Monterey, CA., 1991.

7. Silvasy, S. Jr., *AirLand Battle-Future the Tactical Battlefield*, Military Review, January 91.

8. *FM71-1, The Tank and Mechanized Infantry Company Team*, Department of the Army, 1986.

9. *SIMSCRIPT II.5 Programming Language*, CACI Products Company, La Jolla, CA., 1987.

10. Washburn, A., *Notes on Firing Theory*, Naval Postgraduate School, Monterey, Ca., June 1985.

11. *Encyclopedia of Statistical Terms*, John Wiley & Sons, Inc., New York, NY., 1982.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Deputy Undersecretary of the Army for Operations Research Room 2E621, Pentagon Washington, D.C. 20310	1
4.	Commander U.S. Army Field Artillery School ATTN: ATSF-CCL (MAJ WROTH) Ft. Sill, OK 73503	1
5.	Bell Hall Library U.S. Army Combined Arms School Ft. Leavenworth, KS 66027	1
6.	LTC William Caldwell, Code OR/Cw Department of Operations Research Naval Postgraduate School Monterey, CA 93943-5000	1
6.	Professor Samuel Parry Department of Operations Research Naval Postgraduate School Monterey, CA 93943-5000	1
7.	CPT Malcolm W. Garland 2109 Millerton Ave. Charlotte, N.C. 28208	6

Thesis
G1933 Garland
c.1 KHAFJI.